National Library of Canada
Acquisitions and Bibliographic Services Branch
395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale du Canada
Direction des acquisitions et des services bibliographiques
395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

NOTICE

The quality of this microform is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us an inferior photocopy.

Reproduction in full or in part of this microform is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30, and subsequent amendments.

AVIS

La qualité de cette microforme dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de qualité inférieure.

La reproduction, même partielle, de cette microforme est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30, et ses amendements subséquents.

Canada

UNIVERSITY OF ALBERTA

A Methodology for Designing Local Area ATM Networks

By

Chu-Kiat Lim

©

A thesis
submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree
of Master of Science

Department of Computing Science

Edmonton, Alberta
Fall 1994

National Library
of Canada

Acquisitions and
Bibliographic Services Branch

395 Wellington Street
Ottawa, Ontario
K1A 0N4

Bibliothèque nationale
du Canada

Direction des acquisitions et
des services bibliographiques

395, rue Wellington
Ottawa (Ontario)
K1A 0N4

Your file Votre référence

Our file Notre référence

The author has granted an irrevocable non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of his/her thesis by any means and in any form or format, making this thesis available to interested persons.

L'auteur a accordé une licence irrévocable et non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de sa thèse de quelque manière et sous quelque forme que ce soit pour mettre des exemplaires de cette thèse à la disposition des personnes intéressées.

The author retains ownership of the copyright in his/her thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without his/her permission.

L'auteur conserve la propriété du droit d'auteur qui protège sa thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

ISBN 0-315-95069-2

Canada

Name CHU-KIAT LIM

Dissertation Abstracts International is arranged by broad, general subject categories. Please select the one subject which most nearly describes the content of your dissertation. Enter the corresponding four-digit code in the spaces provided.

COMPUTER SCIENCE
SUBJECT TERM

0984
SUBJECT CODE

U·M·I

Subject Categories

THE HUMANITIES AND SOCIAL SCIENCES

COMMUNICATIONS AND THE ARTS
Architecture 0729
Art History 0377
Cinema 0900
Dance 0378
Fine Arts 0357
Information Science 0723
Journalism 0391
Library Science 0399
Mass Communications 0708
Music 0413
Speech Communication 0459
Theater 0465

EDUCATION
General 0515
Administration 0514
Adult and Continuing 0516
Agricultural 0517
Art 0273
Bilingual and Multicultural 0282
Business 0688
Community College 0275
Curriculum and Instruction 0727
Early Childhood 0518
Elementary 0524
Finance 0277
Guidance and Counseling 0519
Health 0680
Higher 0745
History of 0520
Home Economics 0278
Industrial 0521
Language and Literature 0279
Mathematics 0280
Music 0522
Philosophy of 0998
Physical 0523

Psychology 0525
Reading 0535
Religious 0527
Sciences 0714
Secondary 0533
Social Sciences 0534
Sociology of 0340
Special 0529
Teacher Training 0530
Technology 0710
Tests and Measurements 0288
Vocational 0747

LANGUAGE, LITERATURE AND LINGUISTICS
Language
- General 0679
- Ancient 0289
- Linguistics 0290
- Modern 0291

Literature
- General 0401
- Classical 0294
- Comparative 0295
- Medieval 0297
- Modern 0298
- African 0316
- American 0591
- Asian 0305
- Canadian (English) 0352
- Canadian (French) 0355
- English 0593
- Germanic 0311
- Latin American 0312
- Middle Eastern 0315
- Romance 0313
- Slavic and East European 0314

PHILOSOPHY, RELIGION AND THEOLOGY
Philosophy 0422
Religion
- General 0318
- Biblical Studies 0321
- Clergy 0319
- History of 0320
- Philosophy of 0322

Theology 0469

SOCIAL SCIENCES
American Studies 0323
Anthropology
- Archaeology 0324
- Cultural 0326
- Physical 0327

Business Administration
- General 0310
- Accounting 0272
- Banking 0770
- Management 0454
- Marketing 0338

Canadian Studies 0385
Economics
- General 0501
- Agricultural 0503
- Commerce-Business 0505
- Finance 0508
- History 0509
- Labor 0510
- Theory 0511

Folklore 0358
Geography 0366
Gerontology 0351
History
- General 0578
- Ancient 0579
- Medieval 0581
- Modern 0582
- Black 0328
- African 0331
- Asia, Australia and Oceania 0332
- Canadian 0334
- European 0335
- Latin American 0336
- Middle Eastern 0333
- United States 0337

History of Science 0585
Law 0398
Political Science
- General 0615
- International Law and Relations 0616
- Public Administration 0617

Recreation 0814
Social Work 0452
Sociology
- General 0626
- Criminology and Penology 0627
- Demography 0938
- Ethnic and Racial Studies 0631
- Individual and Family Studies 0628
- Industrial and Labor Relations 0629
- Public and Social Welfare 0630
- Social Structure and Development 0700
- Theory and Methods 0344

Transportation 0709
Urban and Regional Planning 0999
Women's Studies 0453

THE SCIENCES AND ENGINEERING

BIOLOGICAL SCIENCES
Agriculture
- General 0473
- Agronomy 0285
- Animal Culture and Nutrition 0475
- Animal Pathology 0476
- Food Science and Technology 0359
- Forestry and Wildlife 0478
- Plant Culture 0479
- Plant Pathology 0480
- Plant Physiology 0817
- Range Management 0777
- Wood Technology 0746

Biology
- General 0306
- Anatomy 0287
- Biostatistics 0308
- Botany 0309
- Cell 0379
- Ecology 0329
- Entomology 0353
- Genetics 0369
- Limnology 0793
- Microbiology 0410
- Molecular 0307
- Neuroscience 0317
- Oceanography 0416
- Physiology 0433
- Radiation 0821
- Veterinary Science 0778
- Zoology 0472

Biophysics
- General 0786
- Medical 0760

EARTH SCIENCES
Biogeochemistry 0425
Geochemistry 0996
Geodesy 0370
Geology 0372
Geophysics 0373
Hydrology 0388
Mineralogy 0411
Paleobotany 0345
Paleoecology 0426
Paleontology 0418
Paleozoology 0985
Palynology 0427
Physical Geography 0368
Physical Oceanography 0415

HEALTH AND ENVIRONMENTAL SCIENCES
Environmental Sciences 0768
Health Sciences
- General 0566
- Audiology 0300
- Chemotherapy 0992
- Dentistry 0567
- Education 0350
- Hospital Management 0769
- Human Development 0758
- Immunology 0982
- Medicine and Surgery 0564
- Mental Health 0347
- Nursing 0569
- Nutrition 0570
- Obstetrics and Gynecology 0380
- Occupational Health and Therapy 0354
- Ophthalmology 0381
- Pathology 0571
- Pharmacology 0419
- Pharmacy 0572
- Physical Therapy 0382
- Public Health 0573
- Radiology 0574
- Recreation 0575
- Speech Pathology 0460
- Toxicology 0383

Home Economics 0386

PHYSICAL SCIENCES
Pure Sciences
Chemistry
- General 0485
- Agricultural 0749
- Analytical 0486
- Biochemistry 0487
- Inorganic 0488
- Nuclear 0738
- Organic 0490
- Pharmaceutical 0491
- Physical 0494
- Polymer 0495
- Radiation 0754

Mathematics 0405
Physics
- General 0605
- Acoustics 0986
- Astronomy and Astrophysics 0606
- Atmospheric Science 0608
- Atomic 0748
- Electronics and Electricity 0607
- Elementary Particles and High Energy 0798
- Fluid and Plasma 0759
- Molecular 0609
- Nuclear 0610
- Optics 0752
- Radiation 0756
- Solid State 0611

Statistics 0463

Applied Sciences
Applied Mechanics 0346
Computer Science 0984

Engineering
- General 0537
- Aerospace 0538
- Agricultural 0539
- Automotive 0540
- Biomedical 0541
- Chemical 0542
- Civil 0543
- Electronics and Electrical 0544
- Heat and Thermodynamics 0348
- Hydraulic 0545
- Industrial 0546
- Marine 0547
- Materials Science 0794
- Mechanical 0548
- Metallurgy 0743
- Mining 0551
- Nuclear 0552
- Packaging 0549
- Petroleum 0765
- Sanitary and Municipal 0554
- System Science 0790

Geotechnology 0428
Operations Research 0796
Plastics Technology 0795
Textile Technology 0994

PSYCHOLOGY
General 0621
Behavioral 0384
Clinical 0622
Developmental 0620
Experimental 0623
Industrial 0624
Personality 0625
Physiological 0989
Psychobiology 0349
Psychometrics 0632
Social 0451

UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR: *Chu-Kiat Lim*

TITLE OF THESIS: *A Methodology for Designing Local Area ATM Networks*

DEGREE: *Master of Science*

YEAR THIS DEGREE GRANTED: *1994*

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

(Signed) Lim Chu Kiat

Permanent Address:
20, Solok Piggot,
Pulau Pinang,
10450, Malaysia

Date: Oct 4th 1994

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled *A Methodology for Designing Local Area ATM Networks* submitted by *Chu-Kiat Lim* in partial fulfillment of the requirements for the degree of Masters of Science.

Supervisor: J. Harms

Examiner: P. Gburzynski

External: B. Schmuland (Mathematical Sciences)

Chairperson: R. Elio

Date: Oct 4th, 1994

To my parents
Mr. and Mrs. Lim Yoke Kwai,

My brothers
Lim Chu-Keong, Lim Chu-Kim,and Lim Chu-Wai

and
All my friends in Edmonton

Abstract

In a local area Asynchronous Transfer Mode network (LATM), interconnections of switches using high-speed links make up the topology of the network. As the number of variables involved is quite large and the interconnection of these switches is quite complex, a method has to be devised so that a topology that satisfies traffic requirements and is cost effective can be obtained.

This thesis describes a method for designing the topology of a local area ATM network. The method is based on the concept of an ATM pipe. Using these ATM pipes, a logical LATM topology is first created, from which the final physical LATM topology is derived. An advantage of this method is that it produces a topology that satisfies given traffic requirements while trying to minimize the cost of the network. This method consists of two parts. The first part involves the calculation of the ATM pipe sizes. A Multiclass Heterogeneous On/Off Sources Fluid Flow model (MCFF) is used for this calculation. The second part is the derivation of the actual physical topology. The main technique used here is heuristic-based algorithms. The effectiveness of both part one and two are examined, and their results presented.

This method successfully generates network topologies for LATM networks.

Acknowledgements

I would like to extend my greatest gratitude to my supervisor Dr. Janelle Harms, for her guidance and support (especially financially!). Without her unfailing support, this thesis would not be possible. Thank you Janelle.

I am also grateful to the members of my examining committee: Dr. Pawel Gburzynski, Dr B. Schmuland and Dr. Renee Elio for their precious time spent in reviewing this work. I am also indebted to the members of the network group, especially Srinivasan Ramaswamy, Kannan Thiruvengadam, and Xiaofan Guo for their valuable suggestions and support.

I would like to thank all my friends in the Computing Science Department, especially Vikas Nehru (we have been friends for a long time), Sydney Lee, Collins Chien, Kacheong Yeung, Magued Fahim, Lettice Tse (thank you for all the conversations). Special thanks also to Dr. Hong Zhang and Dr. Ahmed Kamal for their encouragements that led to my graduate studies. Also to Prof. Ursula Maydell for her guidance and support in publishing my first paper.

Thank you all.

Contents

1 Introduction **1**

1.1 Network Planning and Design Issues 2

1.1.1 Justification 2

1.1.2 Security 2

1.1.3 Network Architecture 3

1.1.4 Manageability 4

1.1.5 Reliability 4

1.1.6 Cost 5

1.2 Network Planning and Design Life-cycle 6

1.2.1 Long Term Planning and Design 6

1.2.2 Short Term Planning and Design 8

1.3 ATM: The Asynchronous Transfer Mode 9

1.3.1 The ATM Layers 9

1.3.2 Quality of Service 10

1.3.3 ATM Hardware 10

1.3.4 Local Area ATM Networks 11

1.4 Thesis Objective 11

1.4.1 Problem Formulation 12

1.5 Thesis Organization 13

1.6 Summary 14

2 Related Work **15**

2.1 Topology Design using the Steepest Descent Method 15

2.1.1 Analysis 16
2.2 The Mentor Algorithm for Topology Design 17
2.2.1 Analysis 18
2.3 The Simulated Annealing Optimization Technique 19
2.3.1 Analysis 20
2.4 The Add Heuristic and Cluster Heuristic for Topology Design . . 20
2.4.1 Analysis 22
2.5 Summary 22

3 The Pipe Based LATM Topology Design Methodology 24
3.1 Introduction 24
3.2 The ATM Pipe, What is It? 25
3.3 PHASE I: Generating the ATM Pipe Network 28
3.4 The MCFF Fluid-flow Model 29
3.4.1 The Mathematical Analysis of MCFF 31
3.4.2 The Computation of MCFF 39
3.5 Simulation Study 39
3.5.1 Simulation Parameters 41
3.5.2 Class Parameters 41
3.6 Simulation Results 42
3.6.1 Single Class Traffic 45
3.6.2 Effects of Pipe Multiplexing 45
3.7 Limitations of the Verification 49
3.8 Summary 50

4 Phase II: Physical Topology Construction 51
4.1 Introduction 51
4.2 Overview of Phase II 51
4.2.1 Switched ATM Network Requirement 53
4.2.2 The Cost Function 53
4.3 PART I: Initial Topology Construction 54
4.3.1 Choosing which pipes to combine 60
4.4 PART II: Topology Improvement 64

4.4.1 Savings Through Switch Combination 64
4.4.2 Topology Improvement Algorithms 67
4.5 PART III: Topology Location 68
4.6 Experiments for Phase II 70
4.6.1 Parameters 70
4.7 Experimental Results 71
4.7.1 Part I: The Initial Topology Construction 73
4.7.2 Effects of Skewness On the Initial Topology 73
4.7.3 Part II: Topology Improvement 75
4.7.4 Repeated Calculation 76
4.7.5 Part III: Topology Location 76
4.7.6 Algorithm Running Time 83
4.7.7 Compatibility of Part II and Part III 83
4.8 Summary 85

5 Conclusion **87**
5.1 Summary 87
5.2 Conclusions and Future Work 88

A Tucker's Homogeneous On/Off Sources Fluid-Flow Model **90**

Bibliography **94**

List of Figures

1	*The Network Planing and Design Life-cycle*	7
2	*A Conceptual Diagram of an ATM Switch with Output Buffering*	11
1	*The ATM Pipe Concept*	26
2	*A 4 Host ATM Pipe Network*	27
3	*A Corresponding LATM Network*	27
4	*The Fluid-Flow Model*	30
5	*The General Heterogeneous On/Off Sources Markov Chain*	33
6	*Loss Probability vs Pipe Size for Multiclass Traffic (6 sources)*	44
7	*Loss Probability vs Pipe Size for Multiclass Traffic (10 sources)*	44
8	*Loss Probability vs Pipe Size for Single Class Traffic (File Transfer)*	46
9	*Loss Probability vs Pipe Size for Single Class Traffic (CBR Voice)*	46
10	*Loss Probability vs Pipe Size for Single Class Traffic (Video Transfer)*	47
11	*Loss Probability vs Pipe Size for Single Class Traffic (Video Retrieval)*	47
12	*Loss Probability vs Savings (5 sources)*	48
13	*Loss Probability vs Savings (10 sources)*	48
1	*Host 1 on a five host network needs to send traffic to the four other hosts (assuming no traffic to itself)*	56
2	*Host 1 is attached to switch 1 and algorithm attempts to assign the each output pipe to a single link.*	56

3 *The number of links required exceeds the maximum switch size, therefore a pipe combination heuristic is applied to combine some pipes together. A new switch is attached to the link that carries the combined pipe.* 57
4 *Algorithm attempts to assign the output pipes to a single link and succeeds.* 57
5 *The resultant subtopology for source host 1.* 58
6 *The resultant subtopology for destination host 2.* 58
7 *Combining subtopologies together.* 59
8 *A fully combined 5 host initial topology. Source hosts are labeled with numbers preceded with an 'S', destinations hosts with a 'D', and switches with an 'SW'.* 61
9 *Algorithm finds the pipe size (1.9) with the biggest fraction. This pipe is discarded as no other pipe can be combined with this.* . . . 63
10 *Algorithm finds the pipe with the next biggest fraction (0.7). Two other pipes can be combined with this pipe (pipes 4 and 5).* 63
11 *Algorithm chooses pipe 4 as it has the bigger fraction among the two. The two pipes are combined.* 63
12 *Since desired total savings is not reached. Algorithm looks for the biggest combinable pipe.* 63
13 *Only one pipe (pipe 5) can be combined with the chosen pipe (pipe 2), therefore combine them.* 64
14 *A total of 2 savings is achieved. Therefore algorithm exits.* 64
15 *Link capacities reduced when switches that have the same descendants combine.* 65
16 *Links eliminated when switches combine with immediate ancestors or descendants* 66
17 *Reduction of link size and elimination of links as a result of switches combining.* 66
18 *The Initial Topology for the 5 Hosts,Set 2,Skewed Traffic Run* . . 74
19 *The Improved Topology Using the Random Algorithm* 77
20 *The Improved Topology Using the Exhaustive Algorithm* 78

21 *The Improved Topology Using the Neighborhood Algorithm* 79
22 *Comparisons of the Topology Improvement algorithms for a 5 Host Network with Skewed Traffic* 80
23 *Comparisons of the Topology Improvement Algorithms for a 5 Host Network with Non-Skewed Traffic* 80
24 *Comparisons of the Topology Improvement Algorithms for a 15 Host Network with Skewed Traffic* 81
25 *Comparisons of the Topology Improvement Algorithms for a 15 Host Network with Non-Skewed Traffic* 81
26 *Comparisons of the Topology Location Algorithms for a 10 Host Network with Skewed Traffic* 82
27 *Comparisons of the Topology Location Algorithms for a 5 Host Network with Non-Skewed Traffic* 82
28 *Average Execution Time for the Topology Improvement Algorithms (in Seconds)* 84
29 *Average Execution Time for the Topology Location Algorithms (in Milliseconds)* 84

1 *The Talker Activity Model Markov Chain* 90

List of Tables

3.1 *The Four Traffic Classes* 42
3.2 *Set One Multiclass Traffic Mixes (6 sources)* 42
3.3 *Set Two Multiclass Traffic Mixes (10 sources)* 43
3.4 *The Effective Intensities for the Four Classes* 45

4.1 *Pipe Mean and Skewness (5 and 10 Hosts)* 72
4.2 *Pipe Mean and Skewness (15 Hosts)* 72
4.3 *The Initial Topology Cost (The Bigger Cost is in Brackets)* 75
4.4 *Results of the Three Topology Improvement Algorithms Before and After Anchoring* 85

Chapter 1

Introduction

Since the invention of the telephone, the world has taken a great leap forward in communications. At present, millions of miles of communication lines span the entire globe serving billions of people in hundreds of countries. And with the advent of present day technologies in digital computing, the world of communication is witnessing a transition from the old low-bandwidth voice-based networks to high-bandwidth multipurpose all digital networks. No longer are the networks carrying only voice traffic; data, still images, and video as well as other multimedia traffic is becoming a vital and integral part of everyday traffic that is being sent over communication networks. This is evident as the number of voice network users grow at an annual rate of 3% whereas data network users increase at an astonishing 20% annually [37]!

With this explosion in the number of new users and thus new networks, present day communication networks are getting more and more complex as more and more networks of different technologies and functionalities are being deployed. Network planning and designing is no longer an exclusive role to large corporations or telephone service providers, but an important and vital part to all organizations however big or small, that envisage the use of communication networks. Foresight, expertise, and careful planning is indispensable so that the communication network that results satisfies not only present day requirements but future usages and/or expansions as well.

This thesis is about network design, specifically for obtaining the topology for

local area Asynchronous Transfer Mode (LATM) networks. We shall begin with this chapter, by discussing some general topics on network design. Next, a brief overview on Asynchronous Transfer Mode is given. Finally, the problem definition and an objective to this thesis will be presented.

1.1 Network Planning and Design Issues

The role of network design involves various issues. Although every organization is different in terms of size, usages and requirements, several issues are common to many of them. A good reference that discusses these issues can be found in [16]. We present a brief discussion of the network planning and design issues here.

1.1.1 Justification

Before a new network is installed, it is important that the need for such a network is justified. Various facts needs to be considered to justify owning a new network, including costs of hardware, installation, network maintenance technicians and administrator, as well as the forecasted network usage. For example, it might be more economical for certain organizations to lease services or lines from common carrier companies, instead of owning one themselves. This is especially true for organizations that are geographically wide-spread; installing long distance cables and equipment just to serve a single organization may very well be too expensive.

Often alternative solutions exit in lieu of installing a network. A network designer should always forecast the would-be traffic usage for the new network carefully. If the projected traffic is too small, applying alternative solutions might be better than purchasing a new network. For example, for rare transfers of small amounts of data, one can download them with a modem or even transfer them physically in magnetic tapes or diskettes.

1.1.2 Security

As organizations put more and more computers onto their network, unauthorized individuals no longer need to have access physically to a machine to gain access

to restricted information. Information theft may cause serious harm to the well-being of a company, for example, a financial institution. A culprit may log on to a machine from far away, or even out of the country to attempt to gain unauthorized access into the system. It becomes harder to bring the culprits to justice, even if they have been discovered in their act, since they are physically far away.

Therefore, security measurements have to be taken to ensure the privacy of the networks. For example, more stringent access validation systems can be employed, and traffic to and from network gateways can be monitored. Repeated login failures should flag an alert to the system administrators for possible unauthorized entry.

In addition, care has to be exercised in choosing the type of network that is acquired. Some types of networks are more susceptible to unauthorized access than others. As a rule of thumb, networks that broadcasts wirelessly, for example radio networks and satellite links can be eaves-dropped easily, and thus is more susceptible to security problems.

1.1.3 Network Architecture

As there are a wide variety of network technologies in the market, it becomes hard to choose a network architecture to use. Each of the network architecture has advantages and disadvantages of their own. Each of them differs in functionality and price.

As to which architecture is the most suitable, there is no specific answer to this question. Some applications can only run on certain types of network architecture, while others work on a wide range of different networks. For example, constant bit rate voice works best with networks that support synchronous transfers such as the Fiber Distributed Data Interface II Network (FDDI-II); it does not work well with networks that exhibit variable delays (for example, Ethernet).

Another factor to consider is expandability of the network architecture. Some networks can be expanded (scaled) more easily than others with increase demands on traffic. One such network is the switched Asynchronous Transfer Mode (ATM) networks, where the aggregate bandwidth of the network can be expanded easily by adding in new switches and links, as compared to a DQDB network where

the extension of bandwidth can only be done by obtaining another network and connecting these networks by bridges.

1.1.4 Manageability

As networks grow bigger and more complex, the need for a network architecture that is manageable becomes crucial to the everyday operations of the network. A network that is assembled from too many types of equipment from multiple vendors with different standards and formats may become too complex in order to function effectively. Segmentation and assembly of packets may be performed more often than necessary due to the different formats and standards of different equipment. Hence, the increase in overhead, and subsequently the degrading in network performance. A network having too many different types of equipment also needs more staff to maintain the network, since more people need to be trained to handle different types of network components. This represents more cost to maintaining the network.

1.1.5 Reliability

While most networks sold today are relatively reliable compared to a couple of decades back, questions of reliability of networks are still important for certain kinds of networks. In fact, reliability in local networks is becoming a more and more important consideration [25]. As an example, airline flight control equipment requires availability 99.99999% of the time. That means communication networks for flight control equipment has to achieve a downtime of less than 4 seconds per year. Other organizations that rely heavily on the usage of networks may find that reliability of their networks is important. A network failure can possibly result in a loss of thousands or even millions of dollars in business.

Some networks on the market are more reliable than others. For example, FDDI with its dual ring architecture is able to reconfigure the secondary ring in case the primary ring fails. Also, mesh networks can be more reliable than fixed topology networks (ring or star) since a link failure in a mesh network can be easily coped by rerouting the traffic via other paths to the destination. It is up to

the network designer to determine which level of reliability is needed, and hence which network is more suitable.

1.1.6 Cost

Perhaps the one of the most important factors in influencing the decision of purchasing which type of network is cost. It is one of the major issues in network design [28]. The actual cost of acquiring a network can be quite complex. However, the cost can be divided into the following categories:

- **Hardware Cost**. Depending on the technology in use, this can be the cost for switches, multiplexers,concentrators, terminators, diagnostic equipment, copper wire, fiber-optic cables, coaxial cable bridges and gateways, etc.
- **Software Cost**. To have the network operational, certain software has to be installed. For example, communication software, network monitoring software, network administration software, etc.
- **Installation Cost**. Included in this category is the cost related for installing hardware and links. This includes labor cost as well as other costs related to installation. For example, putting up conduits or special encasements for fiber-optic cables, or installing fire-proof material to protect communication lines.
- **Facility Cost**. This category includes the cost for locating hardware or links at premises. For example, a room with special ventilation for locating switches, or wiring closets for link distribution. Sometimes, this cost can be the dominant cost for the network. An example is if rights of way over other peoples property have to be purchased in order to install links/equipment.
- **Maintenance Cost** With the network in place, regular maintenance is required to keep the network functioning effectively. The maintenance cost includes the cost of employing/training maintenance technicians as well as replacements of failed or unfunctional equipment.

1.2 Network Planning and Design Life-cycle

Figure 1 shows a diagram of the life-cycle of network planning and design. This life-cycle can be divided into two parts: long term planning and design, and short term planning and design. Long term planning and design involves planning and designing aspects of the network that exerts long lasting effects to the operations or structure of the network, whereas short term planning and design mainly deals with semi-temporary redesigning or reconfiguration of the network to satisfy short term changing traffic usage requirements.

1.2.1 Long Term Planning and Design

Long term network planning and design starts with forecasting the usage characteristics of the network in the long term. Usage characteristics include what applications will be used, and the type and amount of bandwidth that is required to run these applications as well as any other service requirements associated with these applications. Usage characteristics also include the expected number of users in the long run. Other issues such as reliability, security and manageability are considered as well.

After the long term forecasting or planning stage, the network designer surveys the available technologies or soon to be available technologies (for reasons of future compatibility). Information on prices and functions of each types of technology is collected and considered. Questions to be considered include compatibility between existing standards and equipment and/or future standards and equipment.

With one or a collection of desired technologies in mind, the network designer sets out to design the network. Simple networks such as a token ring network or an ethernet network do not require much designing except for the wiring plan for terminals and peripherals that connect onto the network. Other more complex networks such as switched ATM networks or interconnection of networks require more designing to yield a network with the desired functions. Various tools and techniques such as simulation tools, queueing theory and optimization techniques are applied to assist in the designing process. The focus of this thesis is here,

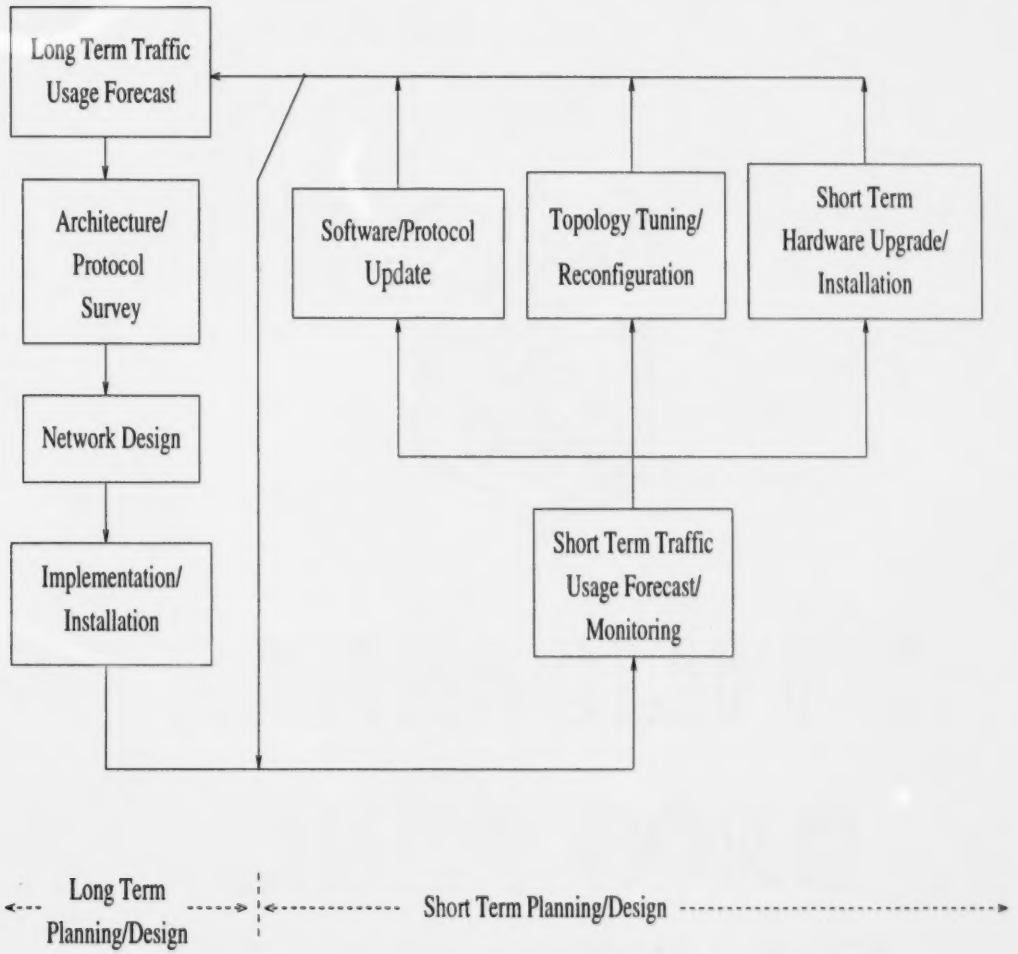

Figure 1: *The Network Planing and Design Life-cycle*

specifically on topology design for switched LATM networks.

Next, the actual purchasing, implementation and installation of the network is done. Some testing is also done in this phase to ensure that the newly installed network is functioning properly.

1.2.2 Short Term Planning and Design

Short term network planning and design, on the other hand is performed in an on-going basis. This process begins with the regular monitoring or short term forecast of network usage statistics. Changes in usage statistics or network requirements are mostly short-lived but the need to fine tune the network persists.

The short term network planning and design process can be divided to the following:

- *Software Update.* By updating or fine-tuning of the software in the topology, some improvements in network performance is achievable. For example, by using software to changing the buffer configuration or cell priority scheme, network efficiency can be improved.

- *Topology Tuning/Reconfiguration* This involves a small change to the network topology, physical or logical. Physical changes include non-major rewiring of the topology. Logical changes include optimization and rerouting traffic to new paths. Some work has been done in this area for ATM networks. Interested readers can refer to [24, 32]

- *Short Term Hardware Upgrade/Installation* Work in this category involves putting in temporary hardware to support short-term increase in traffic or workload. For example, additional multiplexers and concentrators can be installed in to improve link utilization, or adding in additional links to increase bandwidth. Typically, this involves a minor reconfiguration of the topology as well.

1.3 ATM: The Asynchronous Transfer Mode

The Asynchronous Transfer Mode (ATM) is a fast packet switching transfer mode promising great flexibility and support for future bandwidth demanding applications. Since the CCITT (The International Consultative Committee for Telecommunications and Telegraphy) adopted it as the transfer mode for Broadband Integrated Services Digital Networks (B-ISDN) [6], much attention has been focused on research in this area.

B-ISDN is perceived to be an all purpose network [31], promising support for traffic of all bandwidths, including low bandwidth traffic (e.g., voice, telefax, low-speed data, etc.), medium bandwidth traffic (e.g., video telephony, hi-fi sound, etc.) and high bandwidth traffic (e.g.,medical imaging [27], high-quality video distribution, etc).

A characteristic of ATM is its small cell size (53 bytes). Having a small cell size permits statistical multiplexing, hence more efficient utilization of the links. Also, small cell size reduces packetization delay. Another characteristic of ATM is that it is connection-oriented, that is, prior to sending traffic, a virtual channel (VC) is set-up between the source host and the destination host. A virtual channel identifier (VCI) is allocated, and switches route cells based on this VCI. Another identifier is also used along with the VCI. It is the virtual path identifier (VPI). A virtual path is essentially a predefined path in an ATM network. Virtual channels can join in a virtual path so that they can be switched more efficiently as one.

1.3.1 The ATM Layers

The ATM functionality can be divided into three layers [5]:

1. *The Physical Layer.* The first function of the physical layer is to ensure that bits are transmitted correctly onto or received correctly from the physical medium while the second function involves mapping the reconstructed bits to the transmission system used.

2. *The ATM Layer.* This layer is responsible for multiplexing and demultiplexing of cells, cell header addition and extraction, VCI translation and

access flow control.

3. *The ATM Adaption Layer* The main function of the adaptation layer is to provide additional service on top of that already provided by the lower layers. The main aim is to be able to provide different quality of services for different traffic.

1.3.2 Quality of Service

One of the most powerful features of ATM is its ability to provide different quality of services (QOS), that is providing different levels of services for different traffic. For example, voice traffic which is more tolerant to cell loss can specify a higher degree of cell loss, as compared to data traffic. In addition, some traffic requires slightly different services than others. For example, video or voice traffic are sensitive to cell delay jitter, whereas users of data traffic may not care about the jitter in cell delay as long as all the cells arrive within a reasonable time. Basically, QOS can belong to one of three types [3]: deterministic (e.g., maximum delay time to be 500ms), statistical (e.g., cell loss rate to be 10^{-6}) and best effort (e.g, unguaranteed cell delivery).

1.3.3 ATM Hardware

ATM is mainly a switched network, although some proposals have been made for a ring ATM network [34]. Links connect switches with hosts or other switches. ATM links transmit at a certain fixed speed, for example 155Mbps. However, multiples of the link speed can be achieved by combining more than one links together. Each ATM switch contains a number of input ports, where cells arrive, and a number of output ports where cells are sent. At the heart of the switch is the switch fabric that takes care of routing which cell to which output port. Since more than one cell can contend for the same output port, buffers are built in to temporarily store the cells that awaits their turns to be sent. Figure 2 shows a conceptual diagram of an ATM switch with output buffering. For a good introduction to ATM switches, interested readers can refer to [37, 33].

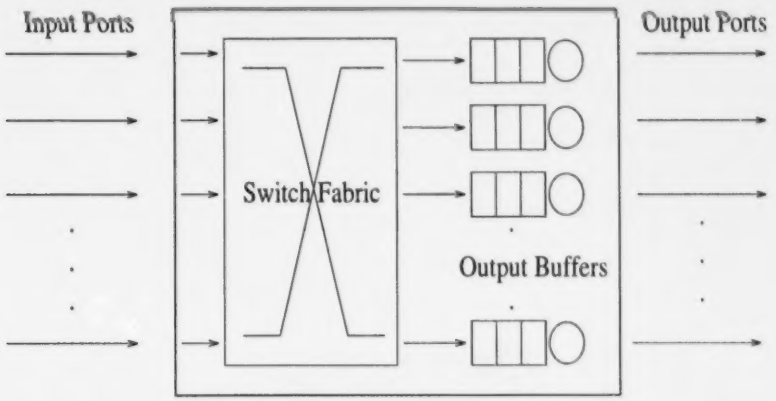

Figure 2: *A Conceptual Diagram of an ATM Switch with Output Buffering*

1.3.4 Local Area ATM Networks

The ATM standard is still evolving. With many aspects of ATM still being unclear it is not surprising that the earliest deployment of ATM networks will be in the local area. As local area networks are usually owned by a single organization, certain functions of the network can be relaxed, for example, congestion control or traffic policing. In addition, the number of users in a local area network environment is smaller, and the amount of traffic is less. Therefore, implementations in a smaller scale are usually done first in a local area environment prior to the metropolitan or wide area networks environments. Because of this, local area ATM Networks are receiving considerable attention, and some first-generation LATM products have already appeared in the market.

1.4 Thesis Objective

Consider the following scenario:

> A network designer is faced with the challenge of designing a local area network. Due to the flexibility and power of ATM he has decided to use ATM as the transfer mode for his local area network. With past usage statistics and careful forecast, he has projected the would-be

traffic requirements (most traffic are multimedia, each with different QOS requirements). He has also obtained information about the cost and characteristics of the LATM switches and links currently on the market. After much consideration, he has decided that the 16x16 and 32x32 switches from company x, and the 64x64 switches from company y are the best buy, and therefore should be used. He has also found out other information on costs, such as labor cost for installing links, and cost for putting in conduits to protect the fiber-optic links. His company has also allocated a few locations specifically for locating the switches for this new network. Under direct orders from the Chief Executive Officer, his network must cost as little as possible while satisfying all traffic requirements. He sits down and begins to ponder on some questions in his mind: How many switches shall he purchase, and which ones? How can he connect the switches so that the desired traffic requirements can be satisfied? And the most important of all, how can the resultant topology be both functional and as cheap as possible? After a while, he realizes he has the answer to the above questions in one of the books on his shelf. He walks to the shelf, reaches out and pulls out the thesis entitled ..... 'A Methodology for Designing Local Area ATM Networks'.

The objective of this thesis is to propose a methodology for solving such a problem. Specifically, this method enables a designer to design a LATM topology that is cost effective while satisfying the requirements of complex ATM traffic.

1.4.1 Problem Formulation

Given the above objective, we can formulate the LATM network design problem as an optimization problem as follows:
Given:

1. Network traffic requirements.
2. Information on usable ATM switches, for example, the switch sizes and costs.

3. Information on ATM transmission links.
4. Locations of user sites.
5. Candidate location sites for switches.
6. The cost associated with installing these switches and links.

Minimize:

1. Total Topology Cost z,
   where z is the cost of switches and links in the topology plus the cost of installing these switches and links.

Over The Variables:

1. Switch number and their corresponding sizes.
2. The location of these switches.
3. The number of links in the network and the interconnection of the switches with these links.
4. The paths that route the traffic.

1.5 Thesis Organization

This thesis is organized as follows. In the following chapter, Chapter 2, a review on some past literature in the area of topology design is given. Several distinctive methods of topology design are outlined there. Next, Chapter 3 presents an outline of our proposed method for LATM topology design. Phase I of our proposed method is included in that chapter. Some results of verifying our Phase I method is presented and discussed. In Chapter 4, the second half of our proposed method, Phase II is presented. The effectiveness of our Phase II method is also examined in this chapter by the discussion of experimental results that we have performed. The conclusions arrived at due to this work are presented in Chapter 5. The pros and cons of our proposed method is discussed along with possible improvements and future works. Finally, appendices are included at the end of this thesis for interested readers.

1.6 Summary

In this chapter, an introduction to network planning and design was given. Various aspects and issues of network planning and design were discussed. We have also given an overview of the Asynchronous Transfer Mode, which is the type of transfer mode used in the networks targeted in this thesis. The scope and objective of this thesis was also presented alongside a formulation of the problem to be solved in this thesis.

Chapter 2

Related Work

While there is little literature for topology design in LATM, much literature exists in the area of topology design in general. In this chapter, we present an overview of several papers that include a wide variety of approaches and techniques for topology design. A brief discussion of each paper is done in relation to their suitability for usage in designing switched LATM networks.

2.1 Topology Design using the Steepest Descent Method

In [11], Gerla, Monteiro and Pazos proposed a method for designing a packet-switched network embedded in a backbone network, and later for reconfiguration of ATM networks [32]. This method involves a concept called "express pipes". An "express pipe" is a logical concatenation of several links that carries traffic from a source node and a destination node. These pipes are derived from the underlying backbone network.

The problem is formulated as an optimization problem, where the variables are the routing of the "express pipes" and the capacities of these pipes, subject to the capacity constraints imposed by the underlying facility network. Gerla, Monteiro and Pazos chose to optimize the average delay of packets in the network. In doing so, they used an M/M/1 model for modeling both the trunk queuing delays and switch delays. By modeling the switches as two nodes of infinite capacity joined by a link of capacity equal to the switch capacity multiplied by the average packet

length, the average delay of the topology is approximated by

$$z = \frac{1}{\lambda} \sum_{u=1}^{\bar{M}} \frac{\bar{f}}{\bar{C}_u - \bar{f}_u} \tag{2.1}$$

where $\bar{M}$ is the number "express pipes", $\bar{f}_u$ is the flow on pipe u, and $\bar{C}_u$ is the capacity of pipe u. This objective function is minimized with the the constraints imposed by the underlying backbone. Using Frank-Wolfe's steepest descent method [29] , the best feasible direction of descent can be found taking the gradient of the objective function, that is

$$\min \bigtriangledown_z (\bar{C}^1, \bar{f}^1) \cdot (\bar{C}, \bar{f}) \tag{2.2}$$

where $(\bar{C}^{K-1}, \bar{f}^{K-1})$ is an initial solution of pipe capacities $\bar{C}$ and pipe flow $\bar{f}$. The gradient [30] of a function is the vector of partial derivatives of that function. By rewriting the equation with the constraints, the problem boils down to a minimum path problem. Gerla, Monteiro and Pazos subsequently applies a modified version of Dijkstra's [2, 14] minimum path algorithm to solve the problem.

2.1.1 Analysis

Using the Steepest Descent optimization technique, Gerla, Monteiro and Pazos in [11] succeed in deriving a logical ATM network (Express Pipe) topology from an underlying ATM backbone network while minimizing the total delay for the network. The delay is calculated from the delays in the links and switches. Although this work deals with ATM topology design, it does not build a network from scratch since they assume that there is an underlying backbone network on which the new topology is derived.

A major drawback of their work is that the links and switches in the network are modeled using M/M/1 queues in order to make the problem more tractable. This is inaccurate for ATM traffic, as ATM traffic is highly complex. The delay in most ATM switches are more accurately modeled by M/D/1, G/D/1 or other queueing models as found in [33, 22, 8, 7, 20, 38, 13, 4, 36]. Another downside to this work is that there is no exact concept of virtual circuits being incorporated into the design process. The traffic from one host can travel to another host via

more than one route. As mentioned in the previous chapter, ATM is a connection oriented protocol, cells belong to a single session are not allowed to travel by multiple routes to the destination. This is necessary so that the cell arrival sequence can be maintained.

In general, this work produces topologies that are optimal. However, due to some of the assumptions that are made, their method is not suitable for our ATM networks.

2.2 The Mentor Algorithm for Topology Design

Kershenbaum, Kermani and Grover in [17] provide a different, more heuristic approach to topology design. Although the solution is not specifically for ATM networks, it is for mesh networks and ATM networks that utilize switches do turn out to be meshes in general. Given the locations of the nodes and the traffic requirements between each source and destination, the algorithm comes up with a minimal cost topology. Unlike the work of Gerla, Monteiro and Pazos in their work in the previous section, who chose to optimize on a performance measure. Kershenbaum, Kermani and Grover try to minimize the cost of the topology instead, assuming that the performance of the network is 'satisfied' by allowing the capacity of the links to be utilized up to a certain level as defined by the topology designer. The reason is that bottlenecks occur and performance deteriorates when links are over-utilized. By keeping the traffic flowing in the links under an acceptable level, reasonably good network performance can be expected.

The first step of Kershenbaum, Kermani and Grover's method is to come up with an initial topology. This topology is an appropriate spanning tree that results in the total link lengths of the network as well as the individual path lengths of destinations being reasonably low. As pointed out by the authors, the networks with the shortest total lengths (minimum spanning trees) will usually result in networks that have longer individual path lengths [16, 17]. while networks with the shortest individual path lengths will usually have a higher shortest total lengths. A special heuristic algorithm has been devised to balance out these two

effects, so that both individual path lengths and shortest total lengths of a network can be kept low at the same time. The heuristic algorithm is a modification of Prim's Minimum Spanning Tree (MST) [2, 14] algorithm and Dijkstra's minimum path algorithm. Given a node as the center of the topology, the algorithm selects candidate nodes to be included into the tree starting at the center node like the Prim's MST algorithm. However, instead of including nodes that have the shortest link l from the edge of the tree, the algorithm includes the node that has the shortest link plus α times the distance from the center node. The center node is determined as the node having the least center of mass

$$M_i = \sum_j c_{ij} w_j \tag{2.3}$$

where c_{ij} is the cost of connecting nodes i and j, and

$$w_j = \sum_k r_{jk} + r_{kj} \tag{2.4}$$

is defined as the total traffic requirements in and out of node j.

The next step involves changing the topology by putting in direct links. A direct link between a source and destination is warranted if the total traffic that flows between this source and destination (link utilization) reaches a minimal level, defined as a portion of a link capacity. If the flows exceed a certain level, more links will be added or alternately a portion of the traffic will be routed by another path. However, since the utilization levels of links depends on the routes of many source-destination pairs, the traffic requirements are sequenced by using a dependency matrix [17].

2.2.1 Analysis

In this work, a heuristic method is applied to successfully create a topology that attempts to minimize both the shortest total lengths and individual path lengths at the same time. The heuristic that they apply is a modification of Prim's MST algorithm and Dijkstra's minimum path algorithm.

While this work manages to generate a topology that is low cost, the cost function in [17] is a little too simple, as it considers the cost of the links as the

only cost. Other cost such as switching components, or installation costs are not considered. Another downside to this work is that the authors assume that the switches in the network can be connected with any number of links, that is the switches have unlimited capacity. In reality, we know that this is impossible. Furthermore, like the previous paper, the traffic from one host to a destination can travel through multiple paths to a destination.

In conclusion, this work utilizes a simple heuristic method to design a network topology. However, since the method is not specifically geared for ATM networks, we can not apply this work to solve our problem.

2.3 The Simulated Annealing Optimization Technique

In [9], Ersoy and Panwar propose a method for designing interconnected LAN-MAN networks. The objective is to find a minimum delay spanning tree topology for interconnected networks. Instead of using M/M/1 to model the delay of the network, Ersoy and Panwar use $M^x/M/1$ queues [12] with batch Poisson Arrivals. The objective function to minimize is

$$T = \frac{1}{\gamma}(\sum_{i=1}^{N} \lambda_i X E[T_i] + \sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}\lambda_{ij} X E[T_{ij}]) \tag{2.5}$$

where $E[T_i]$ is the expected delay in LAN i, $E[T_{ij}]$ is the expected delay in the bridge connecting LAN i and LAN j, and $x_{ij} = 1$ if LAN i and j is connected and equals to zero otherwise. X is mean number of packets in a batch arrival. The expected value of delay for each $M^x/M/1$ queue is

$$E[T] = \frac{X}{\mu(1 - \frac{\lambda X}{\mu})} \tag{2.6}$$

where λ and μ are the arrival rate and service rate respectively.

Ersoy and Panwar proposed the Simulated Annealing Optimization Technique to be used for searching for the solution to this problem. Annealing is a thermal process for obtaining low energy states of a melted solid by slowly cooling the solid. Kirkpatrik, Gelatt and Vecchi [18] first applied an algorithm simulating the annealing process to a combinatorial algorithm. The advantage of the simulated

annealing technique is that it avoids getting caught in a local minima by accepting a cost-increasing transition in the search process. That is, in searching for the global minima, a transition from solution i to j is accepted with probability

$$p = \exp(\frac{-[\text{Cost}(j) - \text{Cost}(i)]}{c}) \tag{2.7}$$

where c is a control parameter that regulates the probability of acceptance. By reducing c towards zero gradually, the probability of accepting a cost-increasing solution decreases. The stopping criterion for simulated annealing is that when the cost does not change after decrementing the control parameter a fixed number of times. This number has to be large enough so that a good part of the search space has been visited.

2.3.1 Analysis

Ersoy and Panwar in their work [9] provide a very elegant method to avoid getting stuck in local minimas using the Simulated Annealing method. This method is generally applicable to any problem that involves searching through a state space to locate a solution. However, like [11] a major downside to their method is the usage of a batch arrival $M^x/M/1$ queue instead of more complex ATM queueing models [33, 22, 8, 7, 20, 38, 13, 4, 36] to model the traffic behavior in the network.

In, short, since this work is mainly for interconnection of MAN LANs with bridges, it cannot be applied directly to solve our problem. However, the Simulated Annealing method is a general search algorithm. Therefore the chances of adapting it into a design problem, including that for ATM networks, is promising.

2.4 The Add Heuristic and Cluster Heuristic for Topology Design

The Add algorithm [16] is a classic algorithm for topology design, initially for attacking the celebrated warehouse location problem. Hsieh, Gerla, McGregor and Eckl adopted and modified this algorithm in [15] for designing the backbone of large packet networks. Specifically, it was used for optimal location of backbone

switches in these networks. Given the location of user sites and relevant traffic requirements for each user site, the problem involves determining the number and location of switches to which user sites connect so that the cost of the backbone topology is minimized. The cost function given by [15] is:

$$D = \sum_i \sum_j \frac{cr_{ij}R(1+b)}{\rho} + N \times F \tag{2.8}$$

where c is the cost per mile per unit bandwidth per month, r_{ij} is the traffic requirement from i to j, b is the line overhead (protocols, etc.), R is an estimate of the average link length, ρ is average utilization, N the number of switches, and F is the average fixed cost per switch.

The modified add algorithm works by weighing the savings achievable by homing a user site to a switch candidate location, instead of homing the user site to an imaginary site which is the center of mass[16] (COM) of the un-homed sites. At each iteration, a new center of mass is calculated, and with it the sites that have positive savings are homed to a new switch which is located at the switch candidate location that achieves the most savings. The homing process during an iteration of the add algorithm will stop when the switch to which the homing is done, is full. The center of mass of two sites[16] is calculated by:

$$\begin{aligned} x_k &= \frac{w_i x_i + w_j x_j}{w_i + w_j} \\ y_k &= \frac{w_i y_i + w_j y_j}{w_i + w_j} \end{aligned} \tag{2.9}$$

where (x_k, y_k) denotes the COM location of sites located at (x_i, y_i) and (x_j, y_j). w_r denotes the traffic weight of site r, which is simply the total traffic that flows in and out of site r.

A second algorithm that is proposed in [15] is the cluster algorithm. The cluster algorithm attempts to minimize the cost of the whole backbone topology by minimizing the local access cost of each site. By partitioning the user sites into N clusters, the sites can be homed to a switch located at the COM of the partition containing the sites. At the beginning of the algorithm, the two sites that are closest together are clustered and located at the COM of the two sites, then at each iteration, the same process is applied to the clusters that are closest

together. To avoid having clusters that are too big or too small, a threshold Z is defined. The merging of clusters is only allowed if the merging does not exceed a threshold Z. At the end of the algorithm, sites that are belong to one cluster will be served by a switch dedicated to that cluster.

2.4.1 Analysis

In this work [15], Hsieh, Gerla, McGregor and Eckl used two very simple heuristics to produce the topology of a network. The Add Heuristic incrementally chooses the best location to home a switch, whereas the Cluster Heuristic attempts to produce a low cost backbone network by minimizing the local access cost for each site. These algorithms are good for their simplicity. However, this is also the downside. This is especially true for the cluster algorithm, where sites are homed to switches without really considering where the traffic needs to be sent.

Unlike the work from the previous sections [11, 9, 17], The authors utilize a more realistic cost function that takes into consideration link and switch costs. However, like [17, 11, 9] a drawback of [15] is that the traffic is too simple compared to current ATM traffic. Traffic is merely specified as some amounts of bandwidth. Another drawback to this work is that the links that they assumed are bidirectional, which is not true for ATM networks.

In general, this work has some good points, specifically the relatively realistic cost function, and simple heuristical algorithms. However, problems do exist for applying this method to an ATM environment.

2.5 Summary

In this chapter, we have presented some research work in the area of topology design each with their own distinctive solution techniques. The solutions presented in these related works are not immediately applicable to the problem that we are pursuing because

- The topology design methods are based on either very simple traffic requirements or none at all. Most papers assume that some r traffic units are

needed for some source-destination pairs without really providing any reason. This is not adequate as traffic that flows in an ATM network consists of different traffic types each with its distinct requirement.

- Traffic behaviors in the networks are approximated using M/M/1 queueing models which is inaccurate in the ATM environment. This is especially undesirable when the objective cost function which is based on the M/M/1 model is optimized to produce the final topology.

- Non-ATM specific hardware is assumed. Most papers assume that switches have infinite capacity, that is, infinite number of links can be connected to the switch. Some methods assume that the links are bidirectional, which is untrue in ATM networks.

- Possible multiple paths exists for ATM traffic. In ATM, to maintain correct sequence of delivery, traffic belonging to one session is required to travel in a single path. Most papers in topology design have no mechanism to incorporate this requirement, hence allowing more than one path to carry traffic of one session to the destination.

Nevertheless, these works are valuable to us, as we have learnt many fundamental ideas from them. Specifically we adopted the idea of a pipe logical network as the starting point for topology design. We also design our heuristic programs to be greedy, similar to the heuristics in the previous work.

Chapter 3

The Pipe Based LATM Topology Design Methodology

3.1 Introduction

Using the problem formulation in Chapter 1 we present here a novel approach for designing a switched local area ATM network topology. This approach is unique compared to past related work in that it produces a low-cost LATM network while guaranteeing certain quality of service requirements much needed by complex multimedia ATM traffic.

There are two phases to our method. Phase I of our method deals with determining the effective multiplexed traffic for each source host-destination host pair. With the knowledge of the effective multiplexed traffic, the quality of service requirements can be guaranteed. The main QOS measurement that we use is cell loss probability. Each source-destination host pair sends traffic down a logical channel, which we termed an ATM pipe. A network of these ATM pipes will be fed into Phase II as input. Phase II generates the actual switched ATM network. It consists of three heuristic-based optimization techniques which attempt to minimize the cost of the final topology.

In Section 3.3, we will describe Phase I of our Pipe-based LATM Design Methodology. This is an analytical method and we call this the Multiclass Heterogeneous On/Off Sources Fluid-flow Model. We will present the details of this

method in Section 3.4. We will also present the results of the simulation study that we have carried out to verify the correctness of this analytical method.

3.2 The ATM Pipe, What is It?

An ATM pipe is a logical, unidirectional pipe that delivers ATM traffic from one host machine to another (Figure 1). It is analogous to a water pipe in everyday life. At one end of this pipe, water (ATM traffic) enters the pipe through the opening and exits at the other end. Since there is only one opening, the amount of fluid in the pipe is directly proportional to the rate of inflow into the pipe. Let's say that congestion in the pipe will cause leakage and hence fluid loss. With knowledge of the pipe size and rate of inflow, the amount of fluid loss will be determinable. Therefore, by using pipes with the right sizes, we can determine the exact fluid loss that may occur.

Using this concept, we can assign an[1] ATM pipe to each source-destination pair to carry traffic. Since the amount of traffic that travels in the pipes is unique to each source-destination pair, the pipe sizes are also different for each source-destination pair. Similarly each source-destination pair may have a different loss rate. An example of a network consisting of ATM pipes is shown in Figure 2. This is a network that consists of 4 hosts, and each host communicates with each other, hence, the fully connected graph.

Before the ATM pipe network can be used, a corresponding physical LATM network has to be designed (Figure 3). Since more than one physical network can be created from a single ATM pipe logical network (using different combinations of switches and wiring of the links, etc.), techniques have to be applied is that the physical network that is created is both functionally equivalent to the logical network and cost effective. In devising this physical LATM network, special attention is made to utilize ATM-specific components (for example, ATM switches and point to point links). This is essential as our aim is to produce a truly ATM network.

In the physical LATM network, the logical ATM pipes are essentially imple-

[1]Theoretically, more than one pipe can be assigned to each source-destination host pair

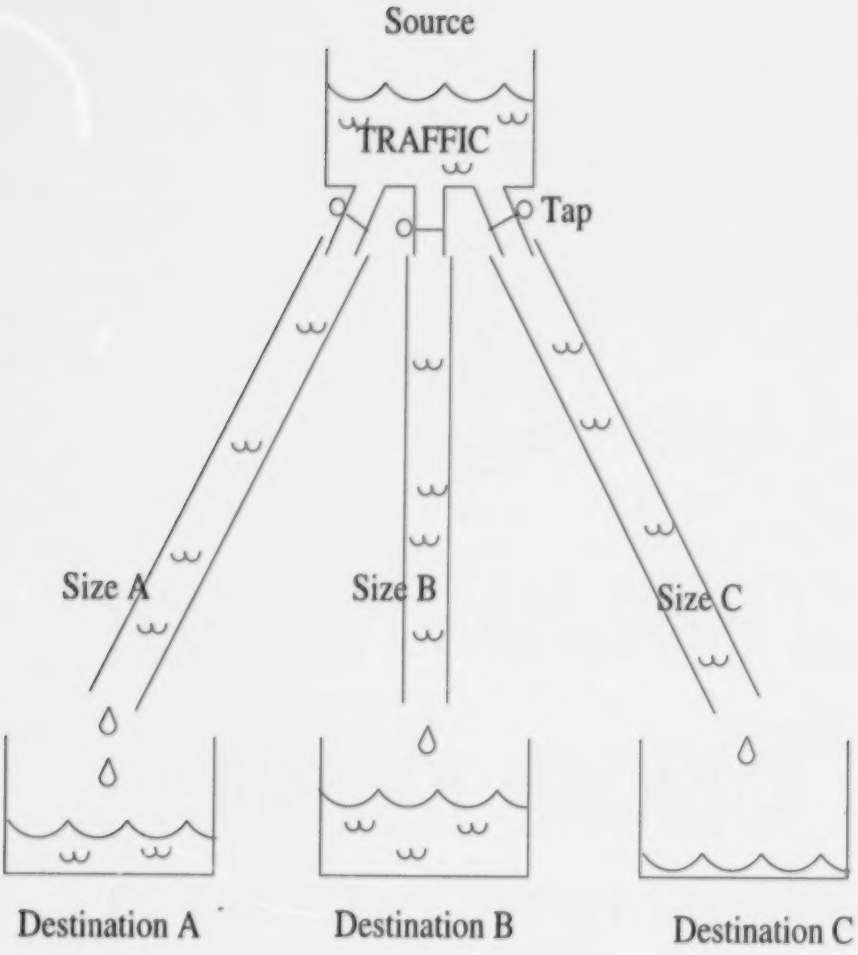

Figure 1: *The ATM Pipe Concept*

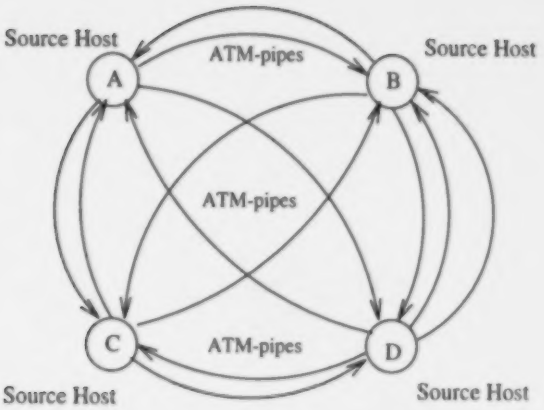

Figure 2: *A 4 Host ATM Pipe Network*

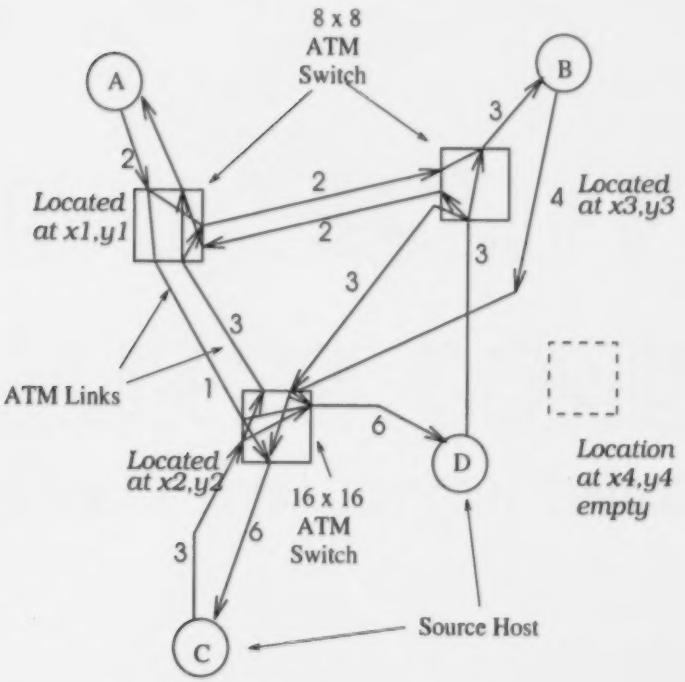

Figure 3: *A Corresponding LATM Network*

mented as permanent virtual paths. With these virtual paths, a set of links or a concatenation of links and switch ports are dedicated to serving each virtual path, so that the effective bandwidth is equivalent to the pipe size. Hence, given the ATM pipe network, the physical topology construction algorithm then decides on the number and dimensions of the switches and the interconnection of these switches with other switches/hosts. The routing of the virtual paths are also computed alongside.

The Power of the ATM Pipe Concept

The power of the ATM pipe concept is its ability to separate the performance aspects and the topology optimization aspects for building a network. While it is not uncommon for topology designers to optimize on performance [16, 9] rather than the cost of hardware, we have chosen the latter as it is hard to optimize on both performance and cost at the same time. By having a pipe between a source and destination, and producing a network based on these pipes, we are essentially producing a 'no wait' network since there is always enough bandwidth (because of the pipes) to handle the traffic, as long as all the hosts do not transmit beyond their pipe sizes. Since all traffic goes in its own pipe, there will essentially be no congestion at all. Furthermore, some other QOS requirements can be implicitly solved. For example, the delay jitter (that is caused by cells' contention for the same output port) will be virtually eliminated. Within the network, delay is limited to fixed delays of switching time and propagation delay in the links.

In this work, we will adopt the traffic loss probability as our main QOS requirement (as our network is a local area network, we assume that the end-to-end delay is satisfied, since it is extremely small). This loss probability will be specified by the network designer when he/she calculates the required pipe sizes.

3.3 PHASE I: Generating the ATM Pipe Network

The ATM pipe Network is simple. It can be represented by a graph $G =< V, E >$, where $V = \{v_i | i = 1, 2, ...N\}$ is the set of vertices, and $E = \{e_j | j = 1, 2, ...M\}$ is the set of directed edges for graph G. Each vertex v_i corresponds to a host

machine, and each edge e_j corresponds to an ATM pipe that flows from a source $s(e_j)$ to a destination $d(e_j)$. Each ATM pipe e_j has a capacity denoted by $c(e_j)$. In practice, graph G will most likely turn out to be a fully connected graph, since all hosts will most likely wish to communicate with all other hosts. Figure 2 is an example of a fully connected ATM pipe network G.

To complicate things a little, the traffic from a host machine to a destination machine is a multiplexed stream of traffic produced by possibly more than one process in that host. Also each stream of traffic generated by these processes are non-identical. In order to produce the right pipe size for the edges, we propose an analytical method to determine the pipe sizes. We call this method the Multiclass Heterogeneous On/Off Fluid Flow Model.

3.4 The MCFF Fluid-flow Model

The Multiclass Heterogeneous On/Off Sources Fluid Flow Model (hereinafter referred to in short form as MCFF) is a burst scale model of traffic generated by multiple sources. It approximates the arrival of data cells as continuous streams of fluid which arrive at fixed rates. The fluid is then 'drained', at a rate determined by the size of the pipe through which these streams of fluid flow[2]. Since the arrival and service rate of this fluid is fixed, the Fluid-Flow Model is also known as the Uniform Arrival and Service (UAS) model.

The UAS Fluid-Flow model was first introduced in a paper by Kosten [20] in 1974, and later improved upon by Anick, Mitra and Sondhi in [1]. Since then, it has gradually gained popularity as a method for modelling traffic in communication networks, including ATM networks. Examples of such efforts can be found in [4, 39, 21, 26, 7].

In 1988, Tucker [39] applied the UAS Fluid-Flow model to model the effects of a speech multiplexer, specifically for obtaining the loss probability as well as the cumulative distribution function of the buffer queue-length. In Tucker's paper, a number of identical speech sources generate data that are fed into a multiplexer. We have adopted and modified Tucker's method to model the effects

[2]Hence, the name Fluid-Flow Model

of multiplexed ATM communication sessions, rather than that of multiplexed short talk-spurts as originally done by Tucker. During a communication session, a source transmits its data at a fixed rate. We call this rate of arrival the intensity of the flow. The major advantage of our method over Tucker's method is that sources can be non-identical, that is, heterogeneous instead of homogeneous. Essentially what that means is that sources can possess different traffic characteristics instead of being uniform. For comparison, a summary of Tucker's method is included in Appendix A.

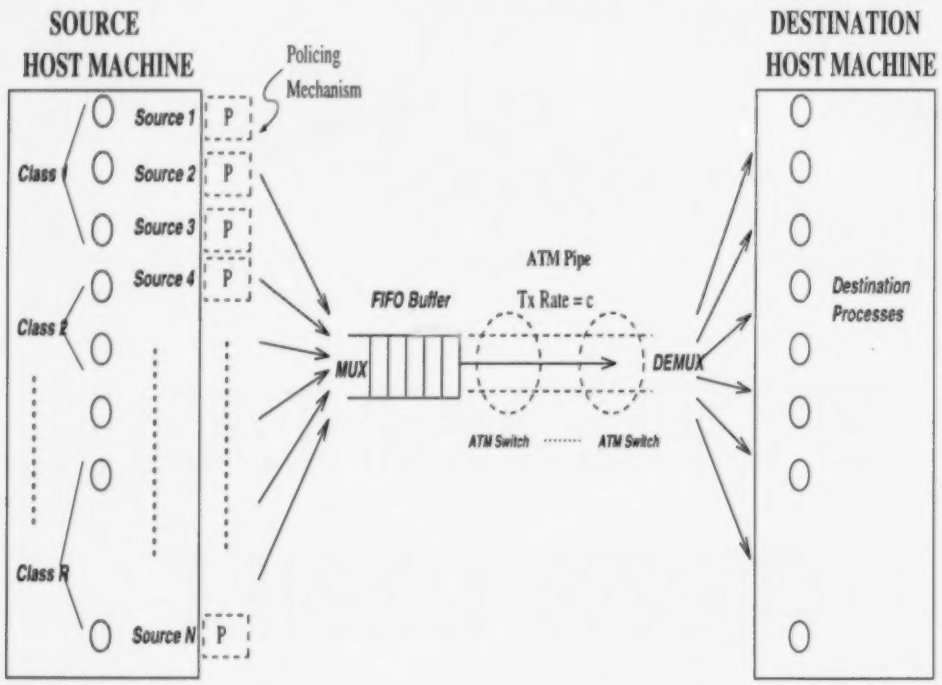

Figure 4: *The Fluid-Flow Model*

Figure 4 shows a conceptual diagram of the MCFF fluid-flow model. Sources at the entry end of the ATM pipe alternate between two states, that is the On state, and the Off state. We assume that the On and Off periods are exponentially distributed. During the On period, traffic flows through a policing mechanism, into a buffer, and then is transmitted onto an ATM pipe en route to the destination

station. The function of these policing mechanisms is to ensure that sources do not violate the maximum bandwidth that they are supposed to transmit.

3.4.1 The Mathematical Analysis of MCFF

The main objective for the analysis of MCFF is to find out the size of the ATM pipe c that is just large enough to satisfy the usages of the traffic sources. With knowledge of the parameters for the sources and the classes to which they belong, and a particular c, we are then able to calculate the loss probability using MCFF. Ideally, we want to be able to calculate the pipe size given a loss probability, however through experimentations, we can find pairs of c and loss to obtain our desired pipe size.

This loss probability is expressed as a quality of service requirement by the pipe designers. Since MCFF approximates the traffic as fluid, there is no concept of packetization. Hence it cannot model the short-term fluctuations in the buffer into which cells arrive.

As shown by Tucker in [39], state equations that form Markov Chains are first derived while taking into account the fluid change in the buffer. At equilibrium probability, these state equations generate differential equations. These differential equations are then solved, and subsequently used to calculate the loss probability of the fluid.

Parameters

Given N traffic sources, each of which belongs to one of R classes such as voice, video and data, we can model the arrival of the traffic from these sources as individual bursts with the following parameters:

$$\text{Mean on period} = \mu_u^{-1} \tag{3.1}$$

$$\text{Mean off period} = \lambda_u^{-1} \tag{3.2}$$

$$\text{Flow intensity} = i_u \tag{3.3}$$

where u is the class to which each these sources belong. Note that a burst is actually a communication session and the intensity of the flow denotes the amount

of bandwidth (in Mbps) dedicated for use by that session.

The MCFF Markov Chain

The Markov Chain [19] has been used widely to study exponentially distributed analytic problems. Figure 5 shows a diagram of the transitions in and out of a state in the Markov Chain used in MCFF. The Markov chain models the On/Off sources as a birth-death process, where a state in the Markov chain describing the number of On sources in each class can move to another state with certain rates, prompted by an arrival (birth) and departure (death) of an On period. Let

$$S = \{s_1, s_2, s_3, \ldots, s_R\} \tag{3.4}$$

be a state S where $s_u, 1 \leq u \leq R$ denotes the number of traffic sources in class u that are in the On period. Relative to S, we define adjacent states to S as

$$\begin{aligned} S(s_u+) &= \{s_1, s_2, \ldots, s_u + 1, \ldots, s_R\}, \quad 1 \leq u \leq R \\ S(s_u-) &= \{s_1, s_2, \ldots, s_u - 1 \ldots, s_R\}, \quad 1 \leq u \leq R \end{aligned} \tag{3.5}$$

Let $r\{S_i, S_j\}$ =transition rate from state S_i to state S_j. Since the inverse of a mean is the rate, the birth and death rates of the Markov Chain can be defined as

$$\begin{aligned} r\{S(s_u+), S\} &= (s_u + 1)\mu_u \\ r\{S(s_u-), S\} &= (w_u - s_u + 1)\lambda_u \end{aligned}$$

or alternately,

$$\begin{aligned} r\{S, S(s_u-)\} &= s_u\mu_u \\ r\{S, S(s_u+)\} &= (w_u - s_u)\lambda_u \end{aligned}$$

where w_u denotes the number of sources belonging to class u. The flow rate out of state S would be the summation of all rates moving out of state S.

$$r^*\{S\} = \sum_{u=1}^{R} \{s_u\mu_u + (w_u - s_u)\lambda_u\}$$

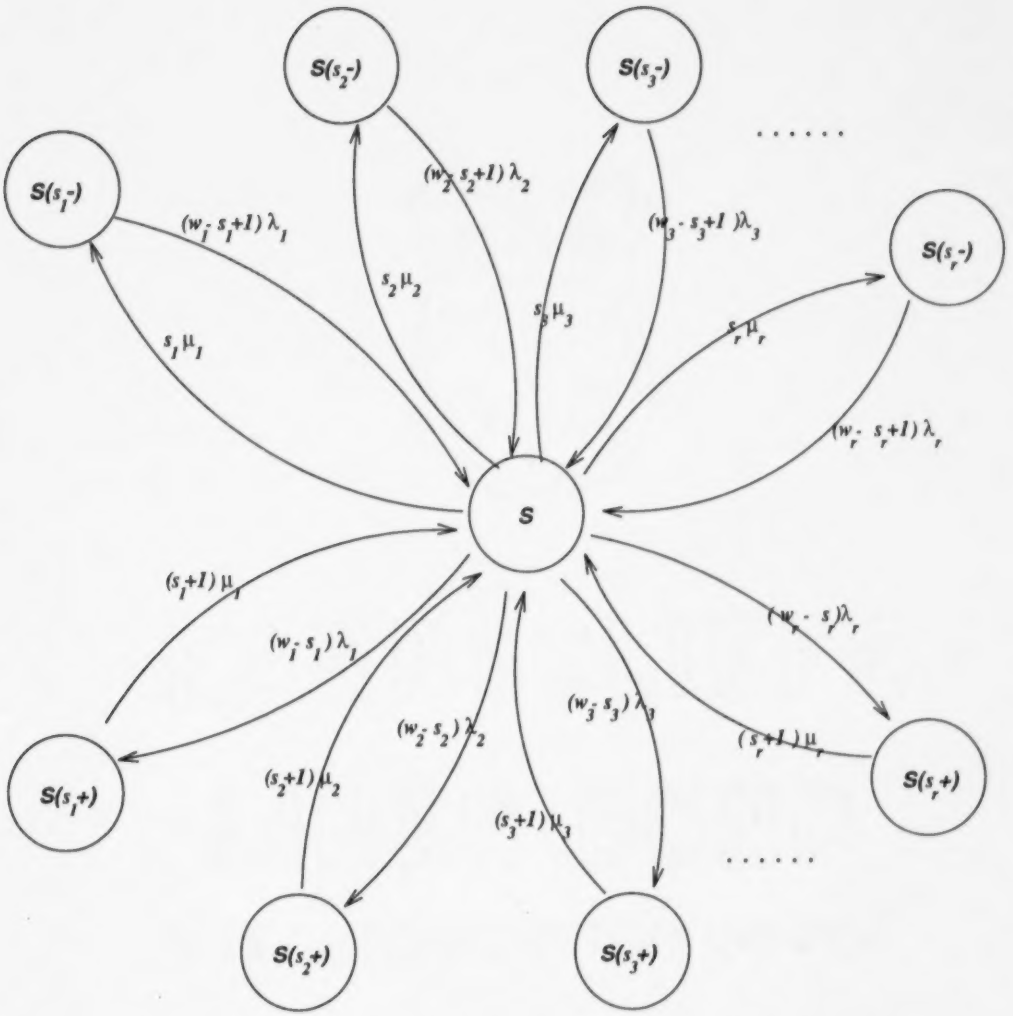

Figure 5: *The General Heterogeneous On/Off Sources Markov Chain*

Note that the birth and death rates prevent S from moving to an invalid state, that is a state where $(s_u - 1) \in S(s_u-)$ is less than zero or where $(s_u + 1) \in S(s_u+)$ is greater than w_u. This is so because the transition rates $r\{S, S(s_u-)\} = 0$ when $s_u = 0$ and $r\{S, S(s_u+)\} = 0$ when $s_u = w_u$.

Deriving the State Equations

As mentioned in [39], the state equations are set by noting the buffer queue behaves in the following manner

1. If the fluid inflow rate > the outflow rate c, the buffer queue increases steadily at a rate of (inflow rate - c). When the queue limit m is reached, the queue will remain at its limit.

2. If the fluid inflow rate < the outflow rate c, the buffer queue decreases steadily at a rate of (c - inflow rate). When the queue reaches empty, it will stay empty.

3. If the fluid inflow = the outflow rate c, the queue level in the buffer does not change.

Define $P_S(t,x), \forall s_u \in S, 0 \leq s_u \leq w_u, 1 \leq u \leq R; t \geq 0; m > x > 0$ be the probability that given that the number of sources in the On state are S, the content of the buffer is not more than x at time t, with a buffer limit of m, we obtain,

$$\begin{aligned} P_S(t+\Delta t, x) \;=\; & \sum_{u=1}^{R} [P_{S(s_u-)}\{t, x - (B(S(s_u-)) - c)\Delta t\} r\{S(s_u-), S)\}\Delta t \\ & + P_{S(s_u+)}\{t, x - (B(S(s_u+)) - c)\Delta t\} r\{S(s_u+), S)\}\Delta t] \\ & + P_S\{t, x - (B(S) - c)\Delta t\}(1 - r^*\{S\}\Delta t) + o(\Delta t) \\ & 0 < x < m \end{aligned} \tag{3.6}$$

where $B(S)$ is the aggregate rate of input fluid. That is

$$B(S) = \sum_{u=1}^{R} s_u(i_u), s_u \in S$$

From (3.6), moving the $P_S\{t, x-(B(S)-c)\Delta t\}$ term to the left hand side, dividing both sides by Δt,

$$\frac{P_S(t+\Delta t,x)}{\Delta t} - \frac{P_S\{t,x-(B(S)-c)\Delta t\}}{\Delta t} =$$
$$\sum_{u=1}^{R}[P_{S(s_u-)}\{t,x-(B(S(s_u-))-c)\Delta t\}r\{S(s_u-),S)\}$$
$$+P_{S(s_u+)}\{t,x-(B(S(s_u+))-c)\Delta t\}r\{S(s_u+),S)\}]$$
$$-P_S\{t,x-(B(S)-c)\Delta t\}r^*\{S\}$$
$$0<x<m \qquad (3.7)$$

approximating the L.H.S. and letting $\Delta t \to 0$, Equation (3.7) becomes

$$\frac{\delta P_S(t,x)}{\delta t}+\frac{(B(S)-c)P_S(t,x)}{\delta x} = \sum_{u=1}^{R}[P_{S(s_u+)}(t,x)r\{S(s_u+),S\}$$
$$+P_{S(s_u-)}(t,x)r\{S(s_u-),S\}]$$
$$-P_S(t,x)r^*\{S\}$$
$$0<x<m \qquad (3.8)$$

At equilibrium probability ($t \to \infty$), the first term on the L.H.S is equal to zero. Defining $F_S(x)$ as the equilibrium probability that with the number of On sources in each class $= s_u, s_u \in S$, the buffer content is less than or equal to x,

$$(B(S)-c)dF_S/dx = \sum_{u=1}^{R}[F_{S(s_u+)}(x)r\{S(s_u+),S\}$$
$$+F_{S(s_u-)}(x)r\{S(s_u-),S\}]$$
$$-F_S(x)r^*\{S\}$$
$$0<x<m \qquad (3.9)$$

State Equations in Matrix Form

Since the number of On sources in class $u, 1 \le u \le R$ can take a value from 0 to w_u, the total number of possible values for S is $z = \prod_{u=1}^{R}(w_u+1)$. Therefore, we have a total of z differential equations. We can write the differential equations in matrix form,

$$\mathbf{D}d\mathbf{F}(x)/dx = \mathbf{M}\mathbf{F}(x) \quad 0<x<m \qquad (3.10)$$

where $\mathbf{D} = \text{diag}(\mathbf{TH} - c)$. $\mathbf{T}$ is defined as a $z \times R$ matrix of all possible values for S. Each Column $\mathbf{T}_{.j}$ in $\mathbf{T}$ corresponds to class j of S. Hence, each row is a possible set S. Later, we shall use $\mathbf{T}_{i.}$ as a value of S interchangeably. We call the matrix $\mathbf{T}$ a Truth Matrix. One example for $\mathbf{T}$ can be

$$\mathbf{T} = \begin{bmatrix} 0 & 0 & 0 & & 0 \\ 1 & 0 & 0 & & 0 \\ 2 & 0 & 0 & & 0 \\ & & \vdots & \cdots & \\ w_1 & 0 & 0 & & 0 \\ 0 & 1 & 0 & & 0 \\ 1 & 1 & 0 & \cdots & 0 \\ & & \vdots & & \\ w_1 & w_2 & w_3 & \cdots & w_R \end{bmatrix} \tag{3.11}$$

$\mathbf{H}$ is an $R \times 1$ column matrix of intensities, i.e. $(i_1, i_2, \ldots, i_R)^T$, and c is the outflow rate.

$\mathbf{M}$, which is evident from equation (3.9) is a $z \times z$ matrix of inflow transition rates, that is

$$\mathbf{M}_{ij} = \begin{cases} r\{S(s_u-), S\}, & \text{if } \mathbf{T}_{i.} = S, \mathbf{T}_{j.} = S(s_u-) \\ r\{S(s_u+), S\}, & \text{if } \mathbf{T}_{i.} = S, \mathbf{T}_{j.} = S(s_u+) \\ -r^*(\mathbf{T}_{i.}), & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$
$$1 \leq u \leq R \tag{3.12}$$

We can then use the eigenvalues and eigenvectors method to solve for $\mathbf{F}$.

Solving the State Equations

Eigenvalues and Eigenvectors An eigenvalue α is said to be a value that when multiplied with a vector $\mathbf{k}$ of size n, produces a result which is the product of an $n \times n$ matrix $\mathbf{A}$ multiplied with the vector $\mathbf{k}$. That is,

$$\alpha \mathbf{k} = \mathbf{A}\mathbf{k} \tag{3.13}$$

By rearranging (3.13) to

$$(\mathbf{A} - \alpha\mathbf{I})\mathbf{k} = 0 \tag{3.14}$$

where $\mathbf{I}$ is an identity matrix of size $n \times n$, we can find values for α and hence $\mathbf{k}$, by solving for the roots of the characteristic polynomial of equation (3.14), which is obtained from solving

$$\det(\mathbf{A} - \alpha\mathbf{I}) = 0 \tag{3.15}$$

The roots of the characteristic polynomial are the eigenvalues for (3.14). With each of these eigenvalue, a vector $\mathbf{k}$ can be computed from equation (3.14). This vector $\mathbf{k}$ is known as the eigenvector corresponding to the eigenvalue from which $\mathbf{k}$ is computed.

We rewrite equation (3.10) as

$$d\mathbf{F}(x)/dx = \mathbf{D}^{-1}\mathbf{M}\mathbf{F}(x) \quad 0 < x < m \tag{3.16}$$

By assuming F to be of the form $e^{\alpha x}\mathbf{k}$, we can obtain the eigenvalues and their corresponding eigenvectors by solving the characteristic equation

$$\det(\mathbf{D}^{-1}\mathbf{M} - \alpha\mathbf{I}) = 0 \tag{3.17}$$

The general solution as shown in [39] will be in the form of

$$\mathbf{F}(x) = a_1 e^{\alpha_1 x}\mathbf{k}_1 + a_2 e^{\alpha_2 x}\mathbf{k}_2 + a_3 e^{\alpha_3 x}\mathbf{k}_3 + \cdots + a_z e^{\alpha_z x}\mathbf{k}_z \quad 0 < x < m \tag{3.18}$$

By setting the proper values for the system of $\mathbf{F}(x)$'s, the coefficients $a_j, 1 \leq j \leq z$ can be found by standard numerical algorithms. These values are defined by setting boundary conditions.

The Boundary Conditions

The contents of the buffer varies according to the rate of input of fluid, that is the rate of inflow of data into the buffer. Let us define $\hat{F}_S$ as the probability that the number of active sources in each class is S, and $\tilde{F}_S(x)$ as the probability that the contents of the buffer is greater than $x, 0 < x < m$, given that the number of

active sources in each class is S. We also define $F_S(x)$ as the probability that the contents of the buffer is less than or equals $x, 0 < x < m$, given that the number of active sources in each class is S. We then get,

$$\hat{F}_S = \tilde{F}_S(x) + F_S(x) \tag{3.19}$$

$\hat{F}_S$ is simply

$$\hat{F}_S = \prod_{j=1}^{R} p_j^{s_j} q_j^{w_j - s_j} \begin{pmatrix} w_j \\ s_j \end{pmatrix}, \quad s_j \in S \tag{3.20}$$

where p_j is the probability that a source belonging to class j is on, and q_j is the probability that it is off, which can be specified as the ratio of the On or Off mean over the summation of the two. In other words

$$\begin{aligned} p_j &= \frac{\lambda_j}{\lambda_j + \mu_j} \\ q_j &= \frac{\mu_j}{\lambda_j + \mu_j} \\ &= 1 - p_j \end{aligned} \tag{3.21}$$

and $\begin{pmatrix} w_j \\ s_j \end{pmatrix}$ denotes w_j choose s_j.

By defining boundary conditions, the set of differential equations can be solved. The boundary conditions are set observing the fact that

1. If the rate of input is greater than the rate of output, that is $(\mathbf{T}_i.\mathbf{H} - c) > 0$ then the contents of the buffer will always be increasing. The buffer in this case will never reach its empty state. Therefore we can set $F_S(0) = 0$.

2. If the rate of input is less than the rate of output, that is $(\mathbf{T}_i.\mathbf{H} - c) < 0$ then the contents of the buffer will always be decreasing. The buffer in this case will never reach its limit. $\tilde{F}_S(x)$ will be $= 0$, hence $F_S(m^-) = \hat{F}_S$. Stirctly speaking, since the state equations does not apply when the buffer is at its limit m, we can set the x to $m^- = \lim_{x \to m}$. Nevertheless, we are able to calculate $F_S(m^-)$ by setting $x = m$ in (3.18).

Calculation of the Loss Probability

With the set of linear differential equations solved, the equilibrium rate of fluid loss can be obtained by simply multiplying the probability of the buffer held at its limit, for all possible combinations of S, with the net rate of inflow to the buffer.

$$\text{Total Loss} = \sum_{j=1}^{z} (\mathbf{T}_j \mathbf{H} - c) \tilde{F}_{\mathbf{T}_j}(x) \tag{3.22}$$

The loss probability is expressed as the ratio of total rate of fluid loss over total rate of fluid inflow. Since each source belonging to class j is in the active state $\lambda_j/(\lambda_j + \mu_j)$ of the time, the total rate of inflow is

$$\text{Total Flow} = \sum_{j=1}^{R} \left(\frac{\lambda_j}{\lambda_j + \mu_j}\right) w_j i_j \tag{3.23}$$

Hence, the loss probability is obtained by

$$\text{Loss Probability} = \frac{\text{Total Loss}}{\text{Total Flow}} \tag{3.24}$$

3.4.2 The Computation of MCFF

The analysis of MCFF was computed on a Sun4 computer using the MATLAB math package. As pointed out by Tucker, the $e^{\alpha_j x}$ term can overflow easily for positive eigenvalues. This can avoided by solving for $a_j e^{\alpha_j}$ instead of just a_j in equation 3.18. This is alright because we need both a_j multiplied with e^{α_j} to calculate $F_S(x)$ in equation 3.18.

An advantage to using MATLAB is that computations can be programmed easily. However, the disadvantage is that we we have no knowledge to the algorithms and implementation of the routines. Thus, results might may be less accurate for extremely large or small numbers.

3.5 Simulation Study

To verify the accuracy of the MCFF mathematical analysis, we have performed a simulation study. In the simulations, the loss probability of the ATM pipes were recorded and compared with those obtained by the MCFF analysis.

We have chosen to model the arrival of ATM traffic in our simulations as continuous streams of fluid with no concept of cell packetization. The reason for doing this is two fold: The first reason is that since we are modelling entire communication sessions as a burst, where on average up to 1.8 Gigabytes of data can be transmitted. In this case, the effects caused by packetization at the physical layer is most likely negligible. The second reason is that we do not wish to assume any particular transmission scheme for the physical layer, since the packetization in these schemes vary extensively. For example, the SONET transmission scheme packs ATM cells into its 810-byte frames whereas TAXI transmits ATM cells raw without additional packetization. In future research, if the effects on the cell-scale as well as burst-scale need to be studied, additional analytic methods can be adapted. For example, Kroner, Eberspacher and Theimer [21] applied the fluid-flow approximation in conjunction with an $M/D/1$-S sub-model while Liao and Mason [26] used the fluid-flow approximation with a $G/D/1$ queuing sub-model for modelling the effects at both the burst and cell levels.

In each simulation run, over 10 million events are generated and the resultant loss probability measured. An event is essentially an arrival of an On or Off period for any of the N sources. We have divided each run into a hundred batches, and the loss probability in each batch is recorded. Since the simulations start with all sources being off and the buffer empty, a warm-up period of 3000 events is passed before any real measurements begin. This is done so that the system reaches a state of equilibrium and the transient effects eliminated. The mean loss probability $\bar{x}$ and its corresponding 95% confidence interval are subsequently calculated with the following formulas:

$$\begin{aligned} \bar{x} &= \frac{\sum x_i}{n} \\ 95\%\ \textit{Confidence Interval} &= \bar{x} \pm z_{1-\alpha/2} \cdot \frac{S_{\bar{x}}}{\sqrt{n}} \\ S_{\bar{x}}^2 &= \frac{\sum (x_i - \bar{x})^2}{n-1} \end{aligned} \tag{3.25}$$

where x_i the loss probability for batch i, $S_{\bar{x}}$ is the standard deviation associated with the mean $\bar{x}$, n is the number of batches which is 100, and for a 95% confidence interval $z_{1-\alpha/2}$ is equals to 1.96.

3.5.1 Simulation Parameters

The parameters involved in the simulations are:

- *Pipe Sizes.* This is the rate that the input traffic are transmitted onto the links, expressed in Megabits per second (Mbps).
- *Buffer Size.* This is the size of the buffer that holds the traffic before being sent out to the network, expressed in seconds[3].
- *Number of Classes and their Related Parameters.* In our case, the number of classes is 4, and the related parameters are the number of sources in a class, the mean On and Off periods in each class (in seconds), and the intensity for associated with a class(in Mbps).

The values for the pipe sizes range from 0.05 Mbps to 34 Mbps depending on the rate of inflow of the traffic. The values for the buffer size range from 0.5 to 3.0. However, since the buffer value directly determines the maximum queuing delay [4] possibly experienced by the traffic, we have chosen to fixed buffer size to 0.5 seconds for most simulations.

3.5.2 Class Parameters

To make our simulations as realistic as possible, we have chosen 4 of the most common types of traffic for ATM networks. They are:

- *File Transfers (FT)*
- *Constant Bit Rate Voice (CV)*
- *Video Telephony (VT)*
- *Video Retrieval (VR)*

[3]The equivalent buffer size in bits can be converted easily by multiplying the buffer size in seconds with the line transmission rate (i.e, the pipe size)

[4]Some traffic especially real-time traffic such as video and voice are sensitive to delays experienced in the transmission. These traffic will be rendered useless if they are kept too long in a buffer. Therefore, this value should not be too large.

The corresponding intensities and On/Off means for the above classes are presented in the following table. These parameters are taken from [35]. Note that the parameters vary greatly from one class to another. For example, the File Transfer class has a relatively short On mean of 1 second compared to the Video Retrieval class with an On mean of 3 minutes. In addition, the Video Transfer class has the highest intensity at 10Mbps, compared to the intensity for the Constant Bit Rate Voice class at 64kbps which is 156 times smaller.

Class Type	*On mean (s)*	*Off mean (s)*	*Intensity (Mbps)*
File Transfer (FT)	1	333.333333	2
Constant Bit Rate Voice (CV)	100	1250	0.064
Video Telephony (VT)	100	5000	10
Video Retrieval (VR)	180	1801.8018	10

Table 3.1: *The Four Traffic Classes*

3.6 Simulation Results

Simulations for multiclass traffic are run. A total of 8 mixes from 2 sets with different number of traffic sources are defined. Mixes in the first set contain a total of 6 traffic sources from the 4 classes, whereas the second set contains a total of 10 traffic sources. The number of sources in each of the 4 classes for set one is shown in Table 3.2, whereas the number of sources in each of the 4 classes for set two is shown in Table 3.3

Class Type	*Mix 1*	*Mix 2*	*Mix 3*	*Mix 4*
File Transfer (FT)	1	1	3	1
Constant Bit Rate Voice (CV)	1	1	1	3
Video Telephony (VT)	1	3	1	1
Video Retrieval (VR)	3	1	1	1

Table 3.2: *Set One Multiclass Traffic Mixes (6 sources)*

As we can see in Tables 3.2 and 3.3, the number of sources in each class are chosen so that one of the classes dominate the traffic. This is done so as to examine the effects these classes have on the loss probability.

Class Type	*Mix 1*	*Mix 2*	*Mix 3*	*Mix 4*
File Transfer (FT)	1	2	6	2
Constant Bit Rate Voice (CV)	1	1	1	6
Video Telephony (VT)	2	6	2	1
Video Retrieval (VR)	6	1	1	1

Table 3.3: *Set Two Multiclass Traffic Mixes (10 sources)*

The results of the simulation runs along the results obtained from the MCFF analysis for set 1 are shown in Figure 6 and set 2 in Figure 7. In these two figures, the traffic loss probability is plotted against the pipe sizes ranging from 6 Mbps to 20 Mbps. The buffer size is set at 5.0 seconds. As we can see, the graphs shows that the results from MCFF is extremely close to that obtained by simulation. All of the points generated with MCFF lie within their corresponding 95% confidence interval obtained from the simulation runs. This means that MCFF is very accurate.

It is interesting to note that in Figure 6 mixes 3 and 4 exhibit almost identical loss probabilities for the same pipe size. While Mix 3 and Mix 4 have the same composition for the VT and VR classes, Mix 3 has 3 sources in the FT class, and 1 in the CV class, whereas Mix 4 has only 1 in the FT class and 3 in the CV class (Table 3.2). With three sources in the FT class generating an intensity of 2 Mbps each, Mix 3 should exhibit more loss for the same pipe sizes as compared to Mix 4 which has three sources in the CV class generating at a mere 64kbps each. The reason lies in the lengths of the On mean and the Off means for the two classes. Although file transfers use up 2 Mbps each, their average session time is much shorter than that for the CBR Voice. In short, if we define the effective intensity of a class as the original intensity multiply by the fraction of the time a source is active, then the two mixes will show similar effective intensity. We show a calculation of this:

$$\textit{Effective Intensity} = \textit{Original Intensity} \times \textit{Fraction of On Time} \quad (3.26)$$

where the fraction of On time is simply the probability of a source being On,

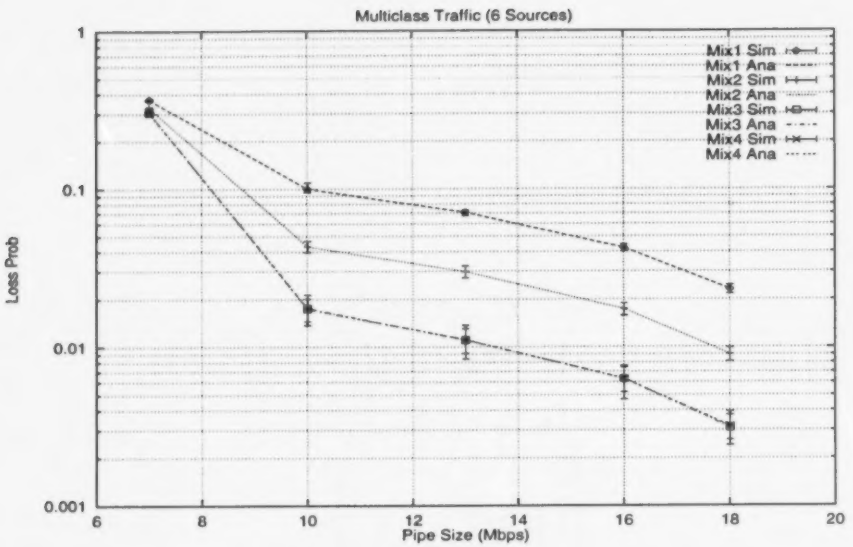

Figure 6: *Loss Probability vs Pipe Size for Multiclass Traffic (6 sources)*

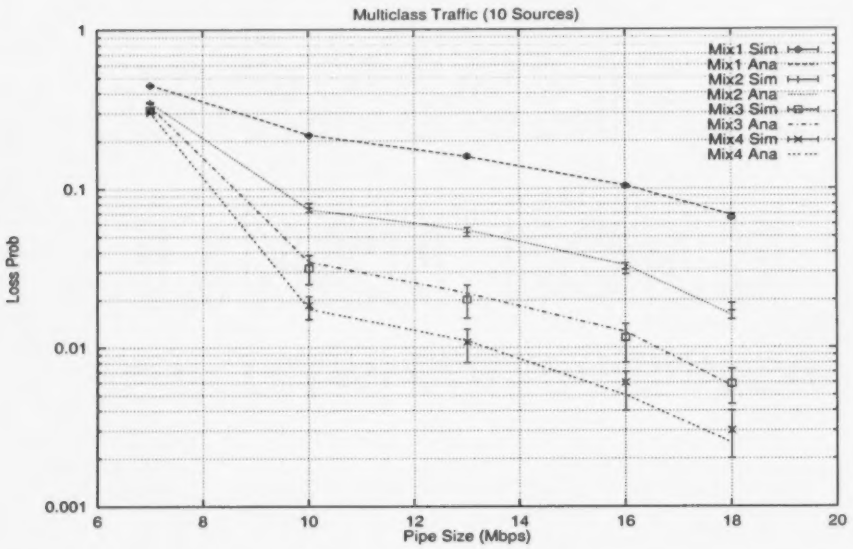

Figure 7: *Loss Probability vs Pipe Size for Multiclass Traffic (10 sources)*

which is

$$Fraction\ of\ On\ Time\ =\ \frac{Mean\ On\ Period}{Mean\ On\ Period + Mean\ Off\ Period} \quad (3.27)$$

The resultant fraction of on time for the FT class is 2.99×10^{-3} and that for the CV class is 0.074. Hence, the effective intensity for FT and CV are 5.98×10^{-3} and 4.74×10^{-3} respectively, which are very close. Table 3.4 shows the effective intensities for all classes. From Tables 3.2, 3.3, 3.4 and Figures 6 and 7, we can see that classes that have higher effective intensities have higher loss probabilities.

Class Type	*Effective Intensities (Mbps)*
File Transfer (FT)	5.982053×10^{-3}
Constant Bit Rate Voice (CV)	4.740740×10^{-3}
Video Telephony (VT)	0.1960784
Video Retrieval (VR)	0.9082644

Table 3.4: *The Effective Intensities for the Four Classes*

3.6.1 Single Class Traffic

To further examine the accuracy of MCFF, we have conducted simulations for the special case of one class, that is the case of homogeneous traffic. For each of the classes FT, CV, VT and VR, two sets of simulation runs are run with different pipe sizes. The first set contains 5 sources and the second contains 10 sources in a class. Again, the traffic loss probabilities in each run are obtained and subsequently plotted with the results from the MCFF computations.

In Figures 8 to 11, the results for homogeneous FT, CV, VT and VR traffic are plotted. Again, we find that the all loss probabilities computed with MCFF lie within their corresponding 95% confidence intervals obtained from the simulations. The results suggests that the MCFF mathematical analysis is indeed very accurate.

3.6.2 Effects of Pipe Multiplexing

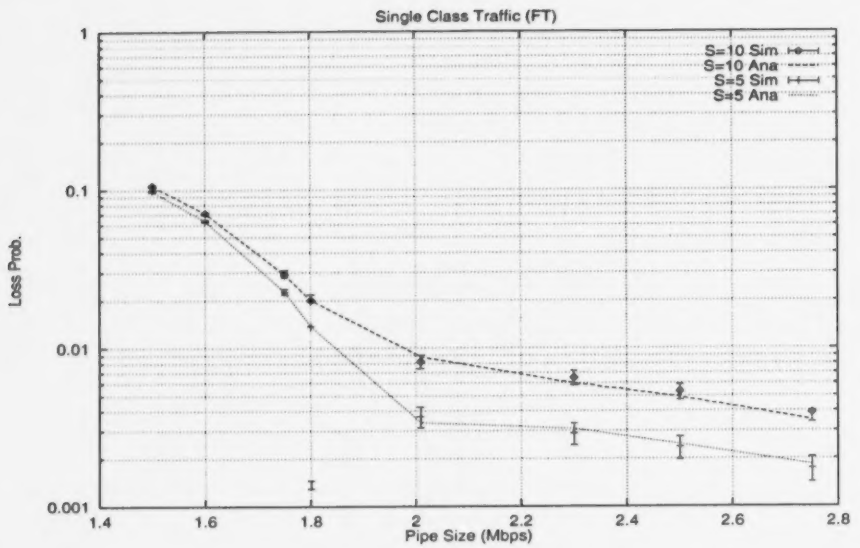

Figure 8: *Loss Probability vs Pipe Size for Single Class Traffic (File Transfer)*

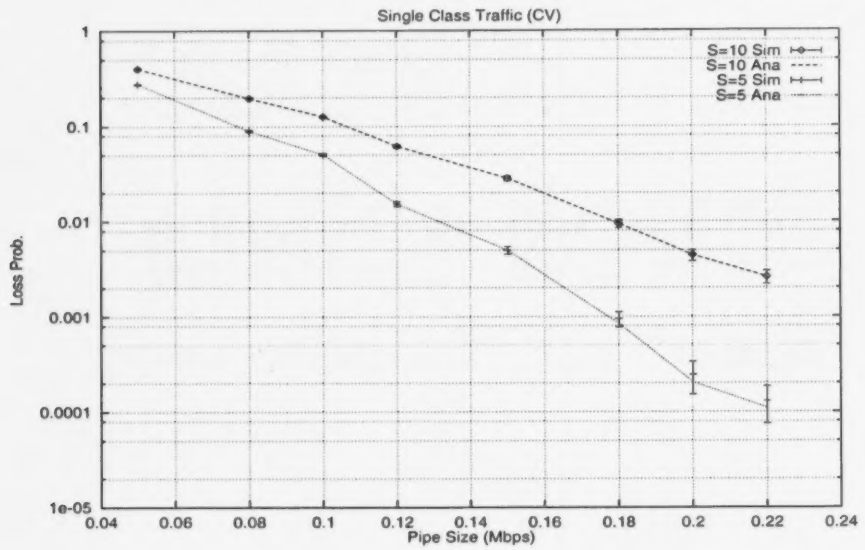

Figure 9: *Loss Probability vs Pipe Size for Single Class Traffic (CBR Voice)*

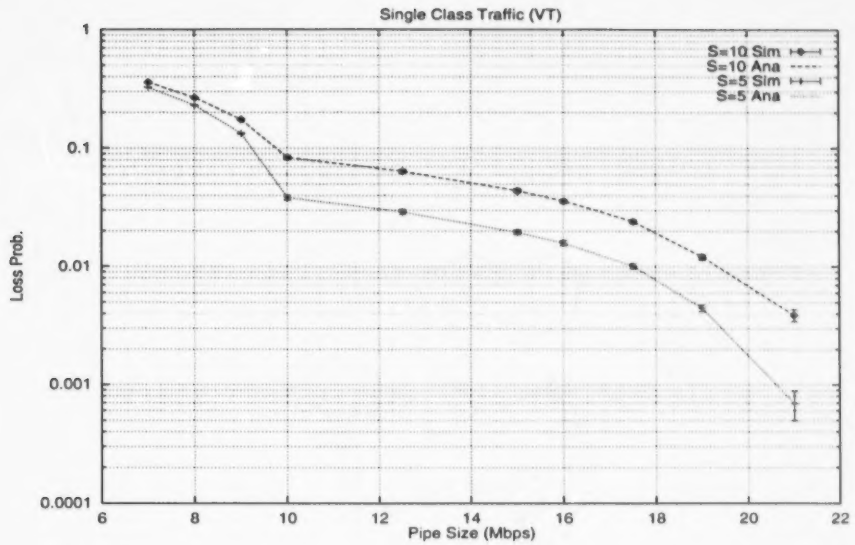

Figure 10: *Loss Probability vs Pipe Size for Single Class Traffic (Video Transfer)*

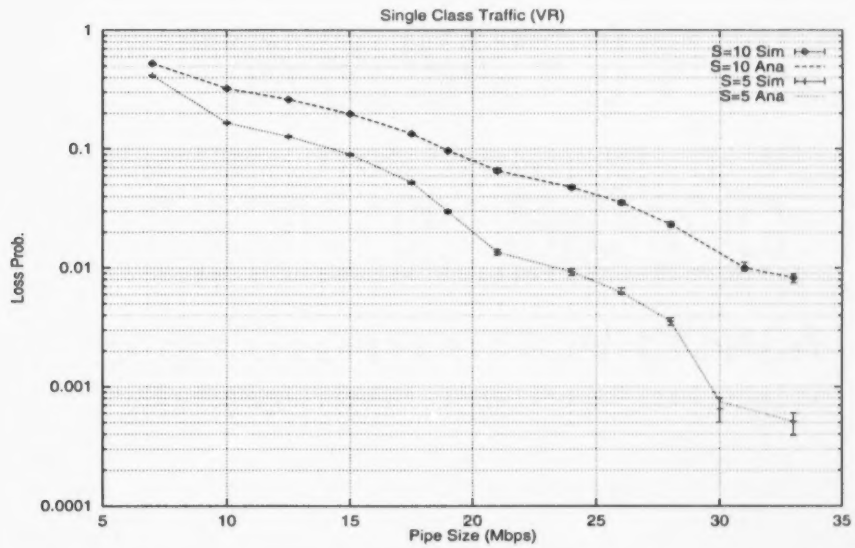

Figure 11: *Loss Probability vs Pipe Size for Single Class Traffic (Video Retrieval)*

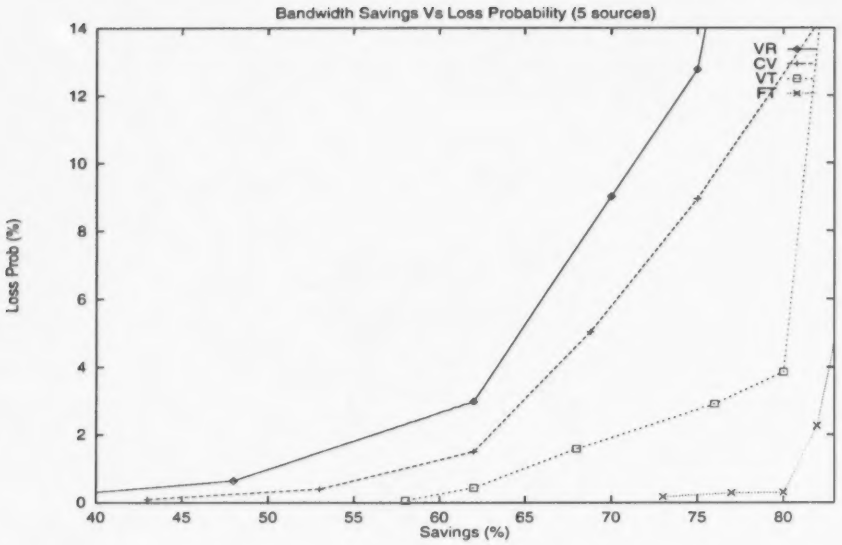

Figure 12: *Loss Probability vs Savings (5 sources)*

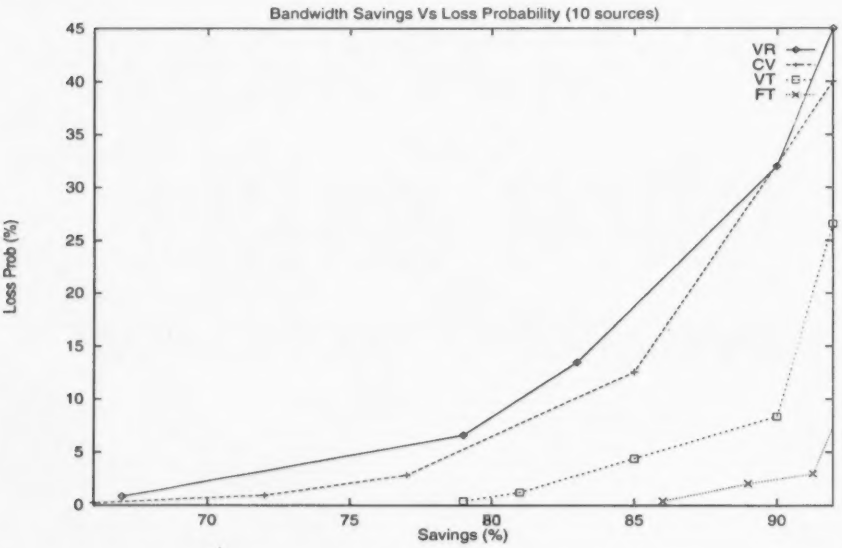

Figure 13: *Loss Probability vs Savings (10 sources)*

With the traffic from the various sources multiplexed onto the ATM pipe, the effective bandwidth will be less than that without multiplexing. This represents savings that is achieved by multiplexing traffic. We define the savings as the ratio between the multiplexed bandwidth i.e. the pipe size, and the unmultiplexed bandwidth, i.e. the aggregate intensities for all sources.

$$\text{Savings from Multiplexing} = \frac{c}{\sum_{u=1}^{R} s_u i_u} \tag{3.28}$$

where c denotes the pipe size, s_u denotes the number of sources in class u, and i_u denotes the intensity for class u. We have plotted the savings against the loss probability for the 5 source runs and the 10 source run for homogeneous traffic. They are shown in Figures 12 and 13.

As is evident from the graphs, the more savings that one desires, the more traffic losses will occur. In addition, the exponential shape of the savings curve suggests that the initial savings achieve the most reductions in traffic loss probabilities. Another interesting observation is that the more sources that share the same pipe, the more savings that one can achieve. This can be seen from the Figures 12 and 13. In Figure 12, where the traffic from 5 sources are multiplexed into a single pipe, for the loss probability of 5 % (VR traffic), the savings achieved is approximately 65%. However, at the same loss probability for the 10 source traffic multiplexing case (Figure 13), the achieved savings an increase of 10% to more at 75%. This can be explained by the individual buffer / shared buffer analogy. Multiplexing traffic sources is analogous to pooling buffers together for sharing. The more buffers that are pooled together the more efficient that the buffers can be utilized. Therefore, what this means is that unless for other reasons, one should always try to multiplex as many sources together in a single pipe, instead of multiplexing them in more than one pipe, as doing this will achieve more savings.

3.7 Limitations of the Verification

As is evident from the graphs that we have verified the loss probabilities of the MCFF model range down to about 10^{-3}, while ATM loss probabilities can be as

low as 10^{-10}. The reason for this discrepancy is twofold: The first reason is that to verify the loss at the range of 10^{-10} will probably require special rare event simulation techniques so that meaningful results can be obtained. The second and probably the more dominant reason is that the we do not have a knowledge of how the computation package that we used, MATLAB, is implemented. For a set of differential equations, MATLAB generates only one eigenvalue with its corresponding eigenvector, while a set of differential equations can have more than one set of eigenvalue and eigenvector. We find that for small loss probabilities, we have to scale the eigenvalues and eigenvectors to obtain the loss probabilities that match the simulation results, hence, defying the whole purpose of verification.

3.8 Summary

In the chapter, we have presented an overview of the Pipe Based LATM Topology Design Methodology. A detailed description of Phase I of this methodology is also described. Phase I essentially deals with the calculation of the sizes of ATM pipes, which are the fundamental building block of our topology design methodology. We presented a analytical method, the Multiclass Heterogeneous Fluid-Flow model (MCFF), for calculating the pipe sizes.

Details of simulation runs for single class and multiclass traffic for verifying the correctness of MCFF were presented. And the results suggest that the MCFF analysis is very accurate, as all points calculated with MCFF lie within the 95% confidence interval of their corresponding simulation results.

Chapter 4

Phase II: Physical Topology Construction

4.1 Introduction

Given that the pipe sizes of each source-destination pair have been computed using the method presented in the previous chapter, and hence the ATM pipe network has been derived, we are set to construct a physical LATM topology based on these pipes. We present a method in this chapter for constructing such a topology.

4.2 Overview of Phase II

Phase II of the topology construction methodology applies a heuristical algorithmic approach to constructing a feasible topology. Heuristics, loosely defined as rule-of-thumbs have long been applied to topology design problems. [16] provides a overview to some heuristics algorithm used in topology design.

The physical topology construction algorithm starts off by generating a feasible topology based on the given pipe requirements. Then, applying heuristic algorithms, we try to improve the design of the initial topology by decreasing the cost of the topology. Note that the initial topology is already a feasible topology since it satisfies the given ATM pipe network. An advantage to this approach

is that the topology is kept feasible throughout the whole process of Phase II. Therefore, no time is wasted on examining topologies that are non-feasible[1]. This dramatically cuts down on execution time.

Phase II can be divided into three parts. They are:

- *Part I: Initial Topology Construction.* In this part, an initial feasible topology is created given the pipe requirements. At this point, only the number, type and interconnections of these switches have been determined. As to the location of these switches, they will be handled by Part III.

- *Part II: Topology Improvement.* In this part, heuristical algorithms are applied to reduce the size of the initial topology, and hence the cost of the topology. Each iteration of the algorithm results in a topology that is feasible. We consider three different algorithms for this part. They are:

  1. *Random.*
  2. *Exhaustive.*
  3. *Neighborhood.*

  Each algorithm will be discussed in detail in Section 4.4.

- *Part III: Topology Location.* The resultant topology from Part II is a feasible topology at the logical level, that is each component of the topology has not been assigned a location in the real world. Since the location of these components directly affect the cost and the performance of the topology, their locations must be assigned in a way that is in some sense optimal. There are two algorithms considered for Topology Location. They are:

  1. *Random.*
  2. *Anchoring.*

  These algorithms will be presented in detail in Section 4.5.

[1]Topologies that do not satisfy the traffic constraints as defined by the given ATM pipe network

4.2.1 Switched ATM Network Requirement

The method that we present in this chapter is specifically for LATM networks, therefore certain assumptions exist so that the topology that is generated with this method is indeed an ATM network. In particular, the main network hardware consists of ATM switches. Links connect these switches with hosts or other switches. Each port on these switches is either an input port or an output port. The traffic that travels on these links goes only in one direction. We also assume that the multiplexed lines originate from or terminate at a switch or host, since if the multiplexed lines originate from or terminate at more than one switch or host, the cell sequence will not be maintained which is a violation of the requirement of ATM. All links in our topology have the same transmission speed, so that no additional buffers need be installed in the switches to prevent cell loss.

4.2.2 The Cost Function

Like [15], we choose the cost of hardware and links as the main costs for our topology. Note that theoretically the cost function can be anything subject to the definition of the network designer.

In this thesis, we have chosen to include the four most common cost for a network topology. That is, the switch cost and the cost for installing the switch, the link cost and the cost for installing the links.

Switch cost and link cost make up the hardware cost for the network. The link costs can include the cost of fiber-optic cables or unshielded twisted pair copper wires depending on the speed of transmission and the technology in use. Costs for installing links can include fiber splicing cost, as well as purchasing and installing conduits or any facilities to hold the links.

For the first two parts of the topology construction method, we defined the following formula for calculating the cost of the topology

$$z = \sum_s N_s C_s^T + \sum_{k=1}^{K} \mathrm{cap}(l_k) D_a C^L + I^L K + I^T \sum_s N_s \qquad (4.1)$$

The first two terms are the total switch and link costs, whereas the last two terms are the installation costs for the switches and links; N_s is the number of switches

of type s, C_s^T is the cost of a single unit of switch type s, cap(l_k) is the capacity of link l_k, C^L is the cost per unit length per unit capacity of a link. I^L and I^T are the cost of installing a link and installing a switch respectively while K is the number of links in the whole topology, and S is the total number of switch types. A link is defined as a connection between a switch and a host machine, or in between switches. Therefore, a link can make up of physically one or more transmission lines. D_a is an approximated average length of a link in the topology. For Parts I and II, an approximation of the link lengths is used instead of the real link lengths since the locations of the switches in the topology will not be set until Part III. We define D_a to be simply the average link distance between hosts and switch candidate locations, and between switch candidate locations. It is given by :

$$D_a = \frac{\sum_h \sum_c d(h,c) + \sum_c \sum_c d(c,c)}{h \times c + c \times c} \tag{4.2}$$

where h is the number of hosts, c is the number of switch candidate locations, $d(i,j)$ is the distance in between switch candidate location i and switch candidate location j. Note that $d(r,r)$ is the distance between switches located at the same switch candidate location. In practice, this distance can never be zero, however, if the distances of $d(i,j), i \neq j$ is comparatively much larger, then we can approximate $d(r,r)$ with 0.

At the end of Part III, since the locations of all switches in the topology have been assigned, we can calculate the cost of the topology by the following rewritten form of equation 4.2:

$$z = \sum_s N_s C_s^T + \sum_{k=1}^{K} \text{cap}(l_k) d_k C^L + I^L K + I^T \sum_s N_s \tag{4.3}$$

where d_k is the real length of the link k.

4.3 PART I: Initial Topology Construction

The objective of the initial topology construction algorithm is to generate a local area ATM topology that satisfies the pipe requirements as given to it by the

network designer. It works by creating an ATM subtopology for each source host and each destination host, and joins all these subtopologies into a larger functional ATM topology. Note that a source host and a destination host can be located on the same host machine, as typically a host machine does not only receive but send information as well. The root of this sub-topology is a source host or a destination host.

The initial topology construction algorithm starts by connecting a source host to a switch with a link. Note that this link can consist of more that one physical link depending on the bandwidth of each single physical link[2]. For example, if the total capacity needed for a link is 305 Mbps and the transmission speed for each physical link is 100 Mbps, then a total of 4 transmission lines giving a total of 400 Mbps is needed to implement this link. Inside each of these link flows the traffic of one or more pipes. The algorithm then tries to assign a separate link to each of the pipes that come out of the switch. If the number of desired output links exceeds the maximum allowable switch size, the algorithm will combine some of the pipes into a single link, attach the link to a new switch and attempt to separate the pipes there. The algorithm will repeat until all pipes are assigned to a single link. Figure 1 to 5 shows how this algorithm works.

Given that a source, say source host 1, needs to transmit traffic to all other hosts in a 5 host network, it sends its traffic in 4 ATM pipes (Figure 1). Each pipe(1,j), $2 \leq j \leq 5$ from source host 1 to destination host j has a size which is described in fractions of a transmission line. The algorithm first attaches a switch to the source host with a link as shown in Figure 2. Since all pipes originating from host 1 travel through this link, the capacity needed for this link is the summation of the capacities of all the pipes that flow in this link. The number of actual transmission lines required is the ceiling of the summation of all pipe sizes. In this example, the capacity required is 2.4, hence the number of transmission lines required is 3.

The algorithm then attempts to assign one link per pipe at the output ports of switch 1 (Figure 2). Since the total number of transmission lines required (summation of the ceiling of each pipe) is 5 which is greater than the largest

[2]to avoid confusion, we shall from now on refer to a physical link as a transmission line

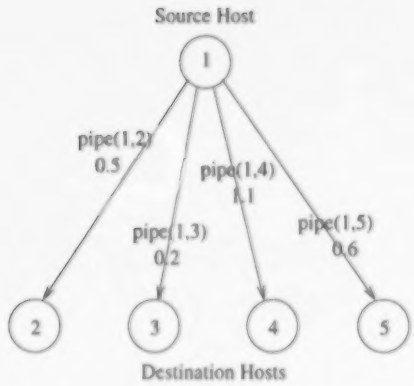

Figure 1: *Host 1 on a five host network needs to send traffic to the four other hosts (assuming no traffic to itself)*

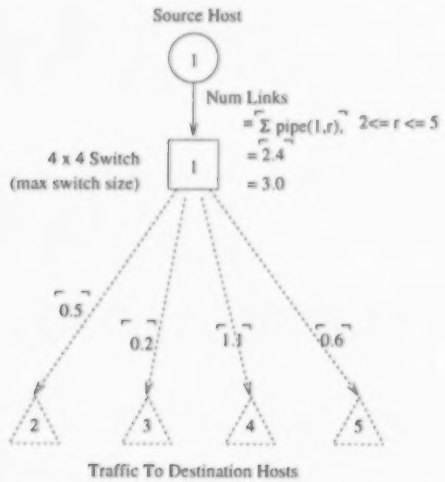

Figure 2: *Host 1 is attached to switch 1 and algorithm attempts to assign the each output pipe to a single link.*

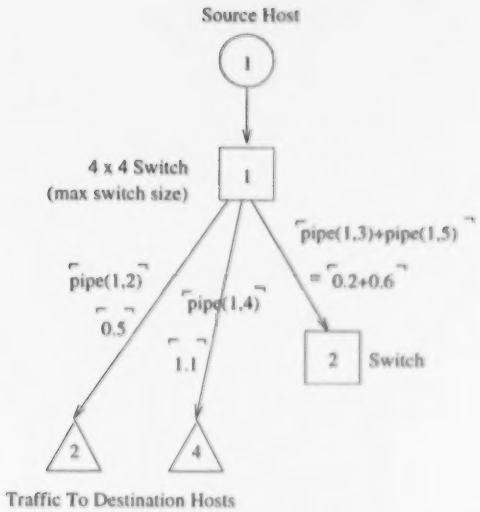

Figure 3: *The number of links required exceeds the maximum switch size, therefore a pipe combination heuristic is applied to combine some pipes together. A new switch is attached to the link that carries the combined pipe.*

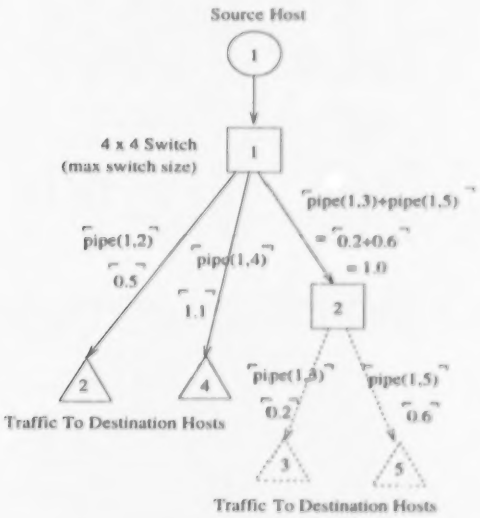

Figure 4: *Algorithm attempts to assign the output pipes to a single link and succeeds.*

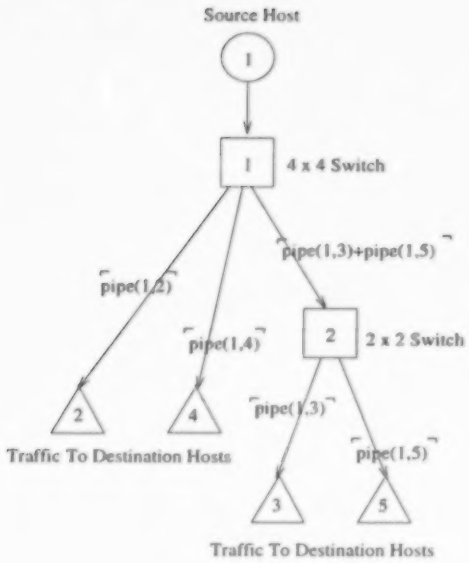

Figure 5: *The resultant subtopology for source host 1.*

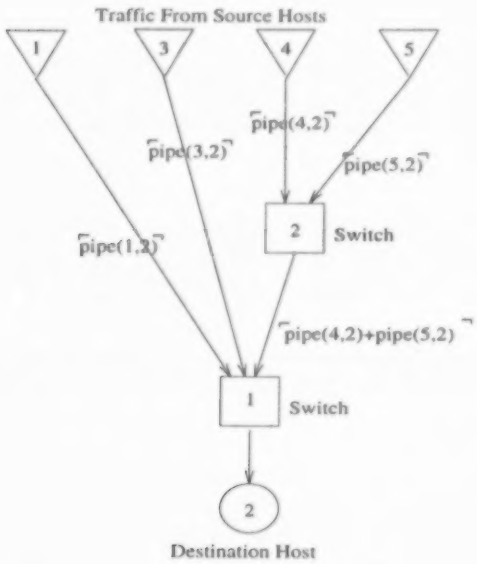

Figure 6: *The resultant subtopology for destination host 2.*

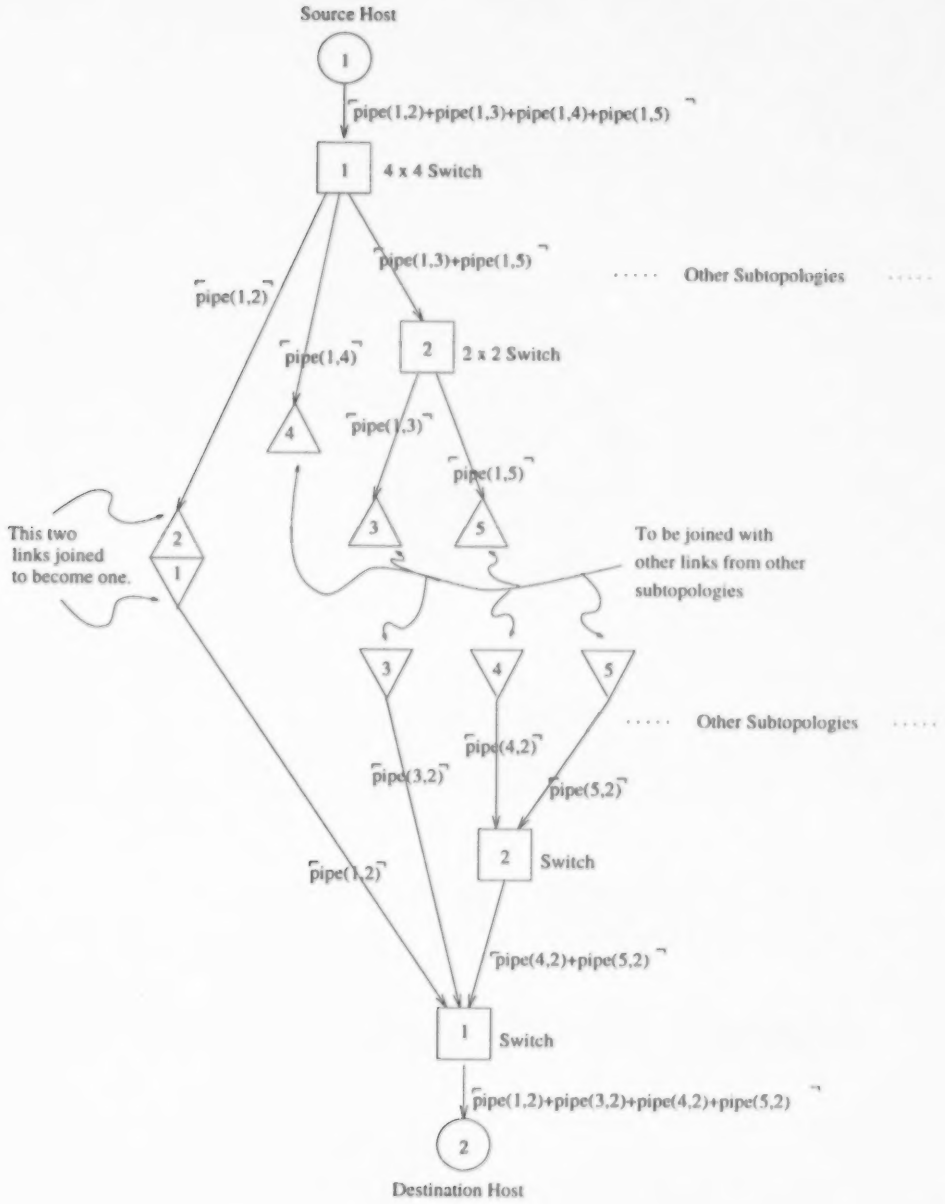

Figure 7: *Combining subtopologies together.*

available switch size (the largest allowable switch has only 4 input ports and 4 output ports), the algorithm combines more than 1 pipe into a link to reduce the number of required lines to fit switch 1. The choice of which pipes to combine is defined by an algorithm which will be described in the next subsection.

By combining pipe(1,3) with pipe(1,5), we reduce the total required output lines to 4, hence satisfying the constraint imposed by switch 1 (Figure 3). A new switch is attached to the link that contains pipe(1,3) and pipe(1,5), as shown in Figure 3. The algorithm then attempts to assign one pipe to a link for pipes (1,3) and (1,5) (Figure 4). Since the total required links is 2, which is smaller than 4, the algorithm succeeds. With only 2 input lines and 2 output lines used, switch 2 can be a 2 x 2 switch instead of a 4 x 4. The resultant subtopology is shown in Figure 5.

The algorithm is repeated for each source host and destination host. A subtopology for destination host 2 is shown in Figure 6. At the leaves of these subtopologies are links containing just one pipe each. Each of these links from a source host subtopology can be matched with a corresponding link from a destination host subtopology. In Figure 5, the leaf links containing pipe (1,2) corresponds to the leaf link containing pipe (1,2) in Figure 6. Therefore, when the topologies to which these links belong combine, these two links will be combined into one, as illustrated in Figure 7. The final combined topology for a 5 host network, can look like the topology in Figure 8. Each source host are labeled with a number preceeded by the letter 'S', each destination host by a 'D', each switch by the letters 'SW', and each link by the letters 'L'.

4.3.1 Choosing which pipes to combine

As mentioned in the previous subsection, if the number of transmission lines required exceeds the maximum allowable switch size, pipes that should normally flow separately in different links are put together in one link, so that the effective number of transmission lines fit within the maximum switch size constraint. We present the algorithm that we used to select which pipes to combine.

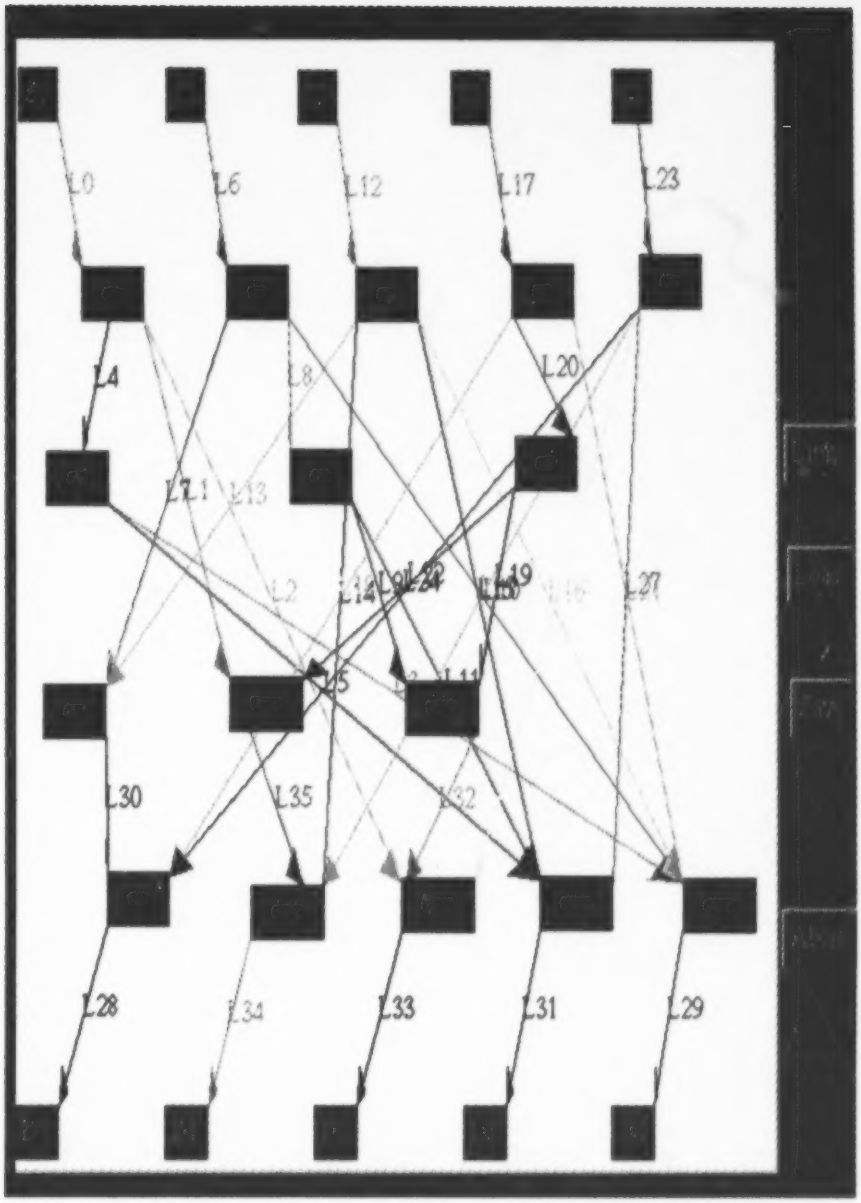

Figure 8: *A fully combined 5 host initial topology. Source hosts are labeled with numbers preceded with an 'S', destinations hosts with a 'D', and switches with an 'SW'.*

The Switch Combination Algorithm

The heuristic employed in this algorithm is that savings in the number of transmission lines can be achieved if two pipes whose fractional part of the pipe size when added together is at most one, are combined into one link. For example, given two pipes a and b, with a having pipe size 1.3 and b 2.5, if the two pipes are put into one link, then the total pipe size is 3.8, and hence the total transmission lines required will be the ceiling of 3.8. That is 4 lines. This represents a savings of one line, because if the two pipes were to put in separate links, then the effective transmission lines required will be the ceiling of 1.3 plus the ceiling of 2.5, which is 5 lines.

The algorithm starts by determining the desired savings needed. Then at each iteration, it chooses two pipes to combine. To increase its chances of combining pipes, the algorithm first chooses a pipe having the biggest fraction. It then looks for another pipe with the biggest fraction that when combined with the first pipe produces a savings of one line. The algorithm then repeats the previous procedure until the desired savings are achieved.

Figures 9 to 14 illustrates how this algorithm works. There are 5 pipes. The total number of transmission lines required is the summation of the ceilings of all the pipe sizes, that is 6 lines. Let's assume that the maximum allowable switch size is 4, a total of 2 savings are needed. The algorithm starts by looking for the pipe that has the biggest fraction. The algorithm finds pipe 1 which has size 1.9 (Figure 9). The algorithm tries to find another pipe that can be combined with pipe 1. Since none can be found, algorithm looks for the pipe that has the next biggest fraction. Pipe 3 which has a size of 0.7 is selected (Figure 10). There are two pipes, pipes 4 and 5 that can be combined with this pipe. Algorithm chooses the pipe 4 that has the bigger fraction and combines the two pipes (Figure 11). A savings of one line is achieved. The algorithm looks for a combinable pipe that has the biggest fraction. Pipe 2 is selected (Figure 12). This time only one pipe (pipe 5) can be combined. Therefore algorithm combines pipe 2 with pipe 5. A total desired savings of 2 lines are achieved. Therefore the algorithm exits.

In essence, this algorithm functions like the First-Fit Decreasing Bin Packing Algorithm [10] that packs to one unit. In general, the number of savings from

Pipes	1	2	3	4	5
Capacity	(1.9)	0.5	0.7	0.3	0.2

Figure 9: *Algorithm finds the pipe size (1.9) with the biggest fraction. This pipe is discarded as no other pipe can be combined with this.*

Pipes	1	2	3	4	5
Capacity	1.9	0.5	(0.7)	0.3	0.2

Figure 10: *Algorithm finds the pipe with the next biggest fraction (0.7). Two other pipes can be combined with this pipe (pipes 4 and 5).*

Pipes	1	2	3 + 4	5
Capacity	1.9	0.5	0.7 0.3	0.2

+

Figure 11: *Algorithm chooses pipe 4 as it has the bigger fraction among the two. The two pipes are combined.*

Pipes	1	2	3 + 4	5
Capacity	1.9	(0.5)	1.0	0.2

Figure 12: *Since desired total savings is not reached. Algorithm looks for the biggest combinable pipe.*

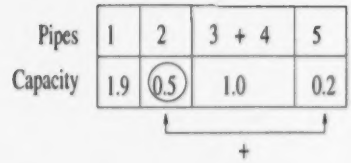

Figure 13: *Only one pipe (pipe 5) can be combined with the chosen pipe (pipe 2), therefore combine them.*

Pipes	1	3 + 4	2 + 5
Capacity	1.9	1.0	0.7

Figure 14: *A total of 2 savings is achieved. Therefore algorithm exits.*

combining of pipes can be further maximized when we pack to u units, where u is a non-zero positive integer. However, we did not implement this and this is left as future improvements.

4.4 PART II: Topology Improvement

The aim of the topology improvement part is to find an equivalent topology that is less expensive than that produced by the initial topology algorithm. An equivalent topology is defined as a topology that satisfies the same pipe specifications as the topology to which it equals.

The topology improvement applies the concept of gradual improvement, that is at each iteration, the algorithm changes the topology according to some heuristics, so that some savings can be achieved. In essence, this concept is similar to the hill-climbing mathematical optimization method [30]. That is, with each iteration, the solution to the cost function gets closer to optimality. Topology changes are done through the intelligent combining of switches Savings are achieved when links rearrange or combine, and the number of switches reduces.

4.4.1 Savings Through Switch Combination

When switches combine, savings are achieved. Specifically

- Total switch cost is reduced as the total number of switches are reduced. Switches that have wasted capacity can be combined so that fewer switches are needed.

- The number of links are reduced when switches that have the same immediate descendants or ancestors combine. This can be seen in Figure 15. Switches 1 and 2 have the same immediate descendants, switches 3 and 4. The total output transmission lines required for switch 1 and 2 is 14. When the two switches 1 and 2 combine, since two of the links out of the four can be combined with the other two, the total required lines have fallen to 12.

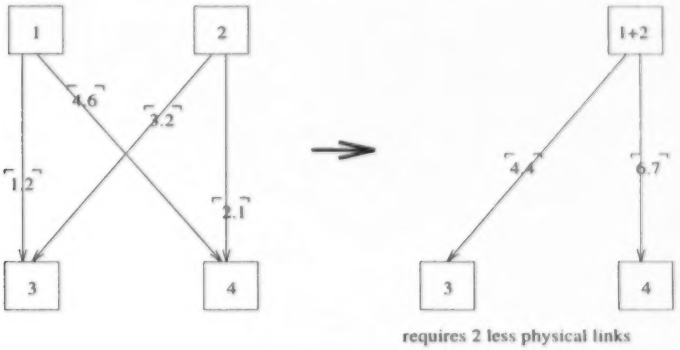

Figure 15: *Link capacities reduced when switches that have the same descendants combine.*

- Links are eliminated when switches combine with their immediate ancestors or descendants. For example, in Figure 16, when switch 1 combines with switch 2, the link carrying pipe a disappears. The total number links required in the topology is reduced.

- Links are eliminated and/or have their capacities reduced when self-loops that occur as a result of switches combining are stripped. This can be seen in Figure 17. When switch 1 combines with switch 3, a self-loop occurs with pipe b and c flowing out of the newly combined switch through switch 2, and back into the newly combined switch. This self loop is stripped and through stripping, links are either reduced in size or totally eliminated.

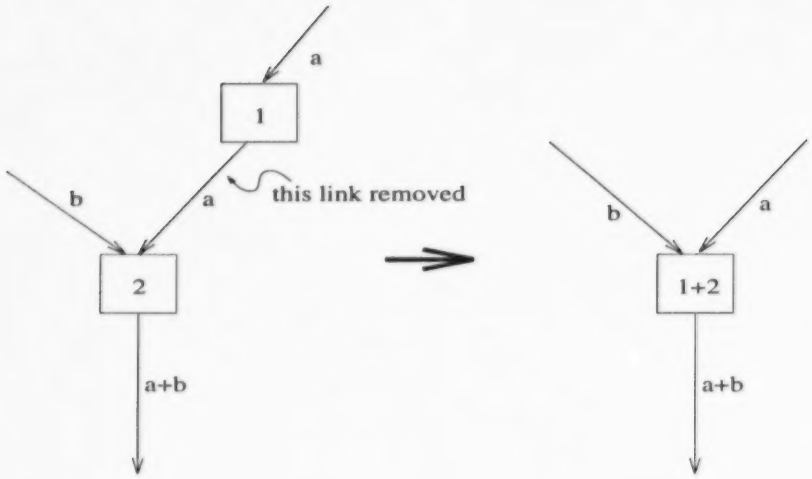

Figure 16: *Links eliminated when switches combine with immediate ancestors or descendants*

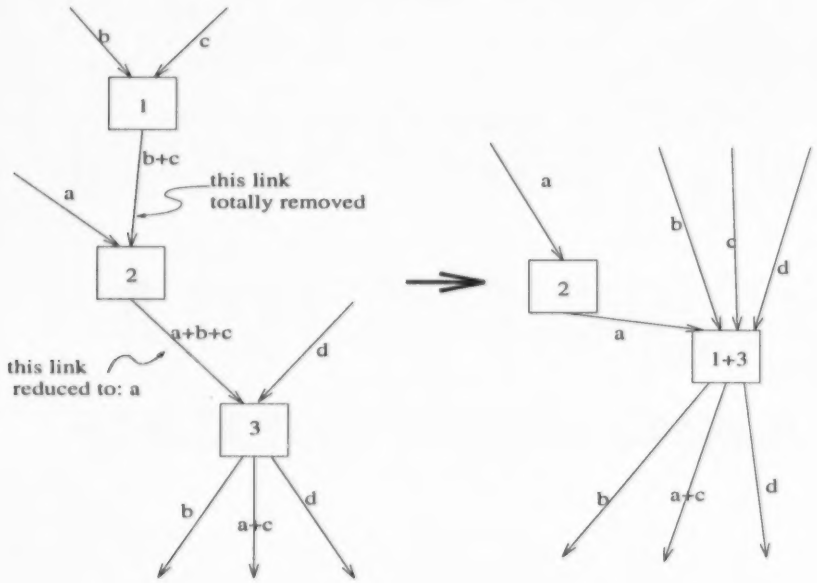

Figure 17: *Reduction of link size and elimination of links as a result of switches combining.*

4.4.2 Topology Improvement Algorithms

We have devised three algorithms to perform topology improvement. These programs each have their own way of choosing which switches to combine and hence the amount of savings achievable. Each algorithm is presented with its pseudo-code.

1. **RANDOM.** The Random improvement algorithm essentially picks two switches randomly. Then it checks to see if the switches are combinable or not. If the switches are combinable, then they will be combined. The algorithm exits when x number of combining attempts results in failure.

```
While(less than x iterations has elapsed since the last
      successful combination}{
           Randomly choose 2 switches in the topology;
           If(the two switches are combinable)
                    Combine them;
}
```

2. **EXHAUSTIVE.** The Exhaustive algorithm in each iteration, finds the two switches that when combined will give the greatest savings. This is a greedy algorithm in that it attempts to achieve the most overall savings by maximizing on the savings in each iteration. This implies more intelligence in the algorithm for building the topology.

```
While(last iterations successfully combines 2 switches){
      Find the savings achievable by all valid combinations of
        any two switches;
      Combine the two switches;
}
```

3. **NEIGHBORHOOD.** The Neighborhood algorithm essentially combines a switch with another from its neighborhood that when combined will give the greatest savings. A neighborhood of a switch is defined as all immediate

ancestors and descendants of the switch, as well as the immediate descendants of these ancestors, and the immediate ancestors of these descendants. A switch can not be combined with itself.

The motivation behind the neighborhood algorithm is that combining switches with its neighbors can most likely result in more links being reduced in size or eliminated (as illustrated in the last section). This facilitates further combination with other switches may result in a cheaper topology.

```
While(less than x iterations has elapsed since the last
       successful combination){
            Randomly choose a switch in the topology;
            Find all the neighbors of this switch as defined
                 by the neighborhood function;
            Find the neighbor that when combined with the
                 switch will give the most savings;
            If(such neighbor is found) combine them;
}
```

4.5 PART III: Topology Location

The topology generated from Part II is a feasible topology. However, this topology is logical as the actual locations of the switches are not assigned yet. The topology location algorithm takes care of allocating the switches to the switch candidate locations. As the locations of this switch directly affect the lengths of the links to which this switch connects, the total cost of the topology is directly affected by the placement of these switches. Two algorithms are devised for performing this task.

- **RANDOM.** This algorithm randomly assigns a switch to a switch candidate locations. This algorithm is run as a basis for comparison.

```
While(not all switches have been assigned){
      Randomly choose a switch candidate location;
```

```
        Assign the switch to that location;
}
```

- **ANCHORING** An anchored switch is a switch that has been assigned a location. At each iteration of the anchoring algorithm, a to-be-anchored switch is chosen randomly. If there are links connecting this switch to other anchored switches, the total lengths of all these links are calculated, with the to-be-anchored switch taking the location of each of the switch candidate locations in turn. Otherwise, pick another unanchored switch. Hence, the location of the switch candidate location that produces the shortest total length will be chosen as the location to which the to-be-anchored switch will be anchored. A host is considered as an anchored switch.

```
While(not all switches have been anchored){
       Randomly choose one unanchored switch;
       If(there are at least one anchored switch
            connected to it){
            For all switch Candidate locations{
                 Find the total distances of all
                    links connecting the unanchored
                    switch with other anchored switch,
                    assuming that the unanchored
                    switch is located at the current
                    switch candidate locations;

            }
            Assign the unanchored switch to the
               location resulting in the shortest
               total distance.
        }
}
```

4.6 Experiments for Phase II

To evaluate the workings of the algorithms proposed for Phase II, we have implemented all of them, and combinations of them (meaning one from Part II and one from Part III) and in a program. With this program, we performed experiments to examine the effectiveness of these algorithms in accomplishing their functions.

4.6.1 Parameters

At the start of each program run, a number of parameters were read in. These parameters are:

- **Number of Hosts.** This is the number of hosts in the network. Values for this parameter are 5,10, or 15.
- **Pipe Sizes.** These are values calculated with the MCFF analytical method presented in the previous chapter. The pipes sizes are expressed in terms of multiples of the line transmission speed. For example, for a 45 Mbps pipe transmitted on a 150 Mbps link, the pipe size will be specified as 45/150=0.30. In the experiments, the smallest pipe size that we used is 0.1 and the largest is 18.3. The number of pipe sizes in a network of N hosts is $N \times (N-1)$, assuming no self traffic.
- **Number of Types of Switches and Their Corresponding Sizes.** Currently, the switches are all symmetrical $n \times n$ types, that is, they have the same number of input and output ports. However, this can be changed easily to accommodate $n \times m$ switches. The Switch sizes that were used in our experiments range from 14 to 200 ports.
- **Switch Costs** As of the time of writing, most switches in the market are listed with the price per port (about \$1000 - \$2000). We have adopted this method of pricing our switches. However since prices change so rapidly, we have decided to omit real dollars as the unit of cost. Rather, we assign the cost for a port as 1 unit and price all other costs (e.g. switch installation, link costs, etc.) relative to that.

- **Switch Installation Cost.** The cost of installing switches. In practice, since this cost differs from case to case (for example, if special facilities such as air-conditioning need to be installed with the switch), we simply fix this value to 0.3.

- **Line Cost.** This is expressed as price per transmission line per meter. Again, this value can vary depending on the type of physical medium (twisted pair, fiber optics, etc) used. Here, we fix this value to 0.005 (for a $2000 per port switch, this works out to about $10 per meter).

- **Link Installation Cost.** Like the switch installation cost and link cost, the link installation cost also varies in real life. Here, we fix this to 0.1. Recall from the previous sections that a link can consist of more than one transmission line and normally they are installed as one. Therefore, we assume that the same link installation cost applies regardless of how many lines there are within the link.

- **Number of Switch Candidate Locations.** These are the places where switches can be installed. The number of switch candidate locations used ranges from to 3 to 8.

- **Distance Between Switches and Hosts or Other Switches.** This is the length of the cable that connects a host with a switch or a switch with another switch. The network designer determines the values for these. Although alternately we can take in actual coordinates of hosts and switch candidate locations as input and let the program calculate the distance, we chose not to do that, since a link often does not take the simplest route to its destination.

4.7 Experimental Results

Experiments of our Phase II program were run on a Silicon Graphics Iris Indigo Workstation. As mentioned before, we have examined 3 sizes of networks, namely,

the 5-host network, 10-host network and 15-host network. For the 5-host and 10-host network, a total of 4 sets of runs were carried out. Each set contains two different classes of pipe parameters with the same mean pipe sizes. The pipe sizes in one class are skewed and the other are not skewed. This also applies for the 15-host network. The only difference is that instead of 4 sets, only 2 sets were run.

We calculate the skewness of the pipes with the following formula which was taken from [23]

$$v = \frac{E[(p_j - \bar{p})^3]}{(S^2)^{3/2}} \tag{4.4}$$

where p_j is a pipe size, $\bar{p}$ is the mean pipe size and S^2 is the variance of the pipe sizes. The pipe mean and skew values for the experiments are listed in Tables 4.1 and 4.2. Note that the smallest pipe mean is 0.5 and the largest is 3.0, whereas the smallest skew value is 2.68 and the largest is 4.593.

Set	*5 Hosts*		*10 Hosts*	
1.	Mean: 1.5 Skew: 3.112	Mean: 1.5 Skew: 0.0	Mean: 1.5 Skew: 3.217	Mean: 1.5 Skew: 0.0
2.	Mean: 3.0 Skew: 3.574	Mean: 3.0 Skew: 0.0	Mean: 3.5 Skew: 3.21	Mean: 3.5 Skew: 0.0
3.	Mean: 2.0 Skew: 2.68	Mean: 2.0 Skew: 0.0	Mean: 2.0 Skew: 2.808	Mean: 2.0 Skew: 0.0
4.	Mean: 0.5 Skew: 3.091	Mean: 0.5 Skew: 0.0	Mean: 0.5 Skew: 4.593	Mean: 0.5 Skew: 0.0

Table 4.1: *Pipe Mean and Skewness (5 and 10 Hosts)*

Set	*15 Hosts*	
1.	Mean: 1.5 Skew: 3.117	Mean: 1.5 Skew: 0.0
2.	Mean: 3.0 Skew: 3.542	Mean: 3.0 Skew: 0.0

Table 4.2: *Pipe Mean and Skewness (15 Hosts)*

4.7.1 Part I: The Initial Topology Construction

In each run, we recorded the cost of the initial topology. We find that in general, the more hosts there are in a network, the bigger the initial topology will turn out to be. This is so because more links and switches will result since the initial topology construction algorithm assigns a switch to each source host and destination host by default regardless of the input pipe sizes.

Figure 18 shows an example of an initial topology generated from one of our runs, that is the 5-host, Set 2, Skewed Pipe run. Each source host and each destination host is connected to a switch. Note that switch 4, which is not attached to any host, is part of the subtopology belonging to source host number 3. It is created because the total needed lines (for the output of switch 3) exceeded the maximum switch size.

At the conclusion of each experimental run, we have also verified links and paths to make sure that the initial topology is a feasible topology. That is, it contains all pipes that were given as the input, and that the maximum number of lines required for a switch does not exceed the maximum switch size. We find that the topologies that were generated by the initial topology algorithm are indeed feasible. As our algorithm is designed to maintain the feasibility of the topology, this checking is done mostly as a check on the implementation,

4.7.2 Effects of Skewness On the Initial Topology

Table 4.3 shows the initial topology costs gathered for the 5 and 10 host runs. In there, the results of both skewed and non-skewed costs are shown. The bigger cost of the two are enclosed in brackets.

From the table, we can see that number of times that the results from the skewed runs exceeds the results from the non-skewed runs are exactly the same as the other way round. This suggests that the skewness of the pipe sizes does not appear to have a direct influence over the cost of the initial topology. This is contrary to our intuition. One explanation could be that the difference in pipe sizes (hence the skewness) is not really the main factor that influences the number of switches and links that were used in the topology. The main factor is

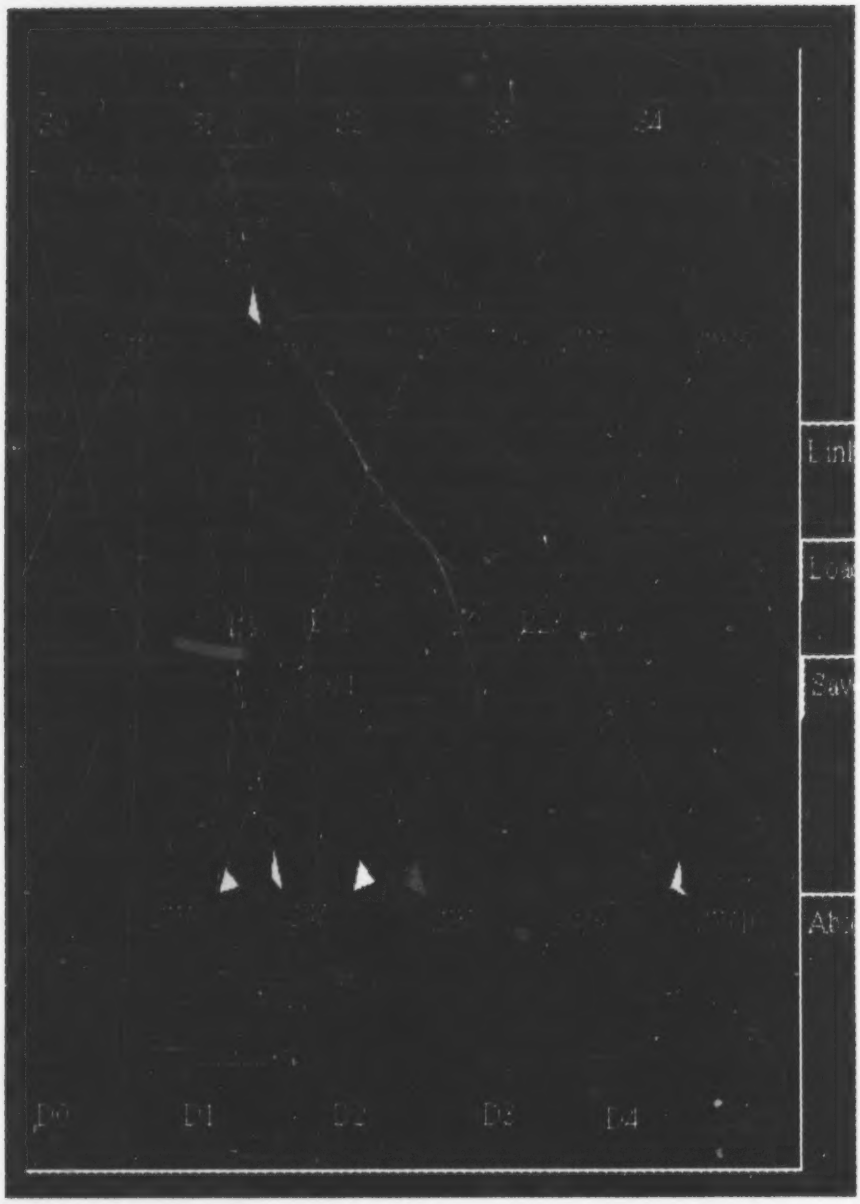

Figure 18: *The Initial Topology for the 5 Hosts,Set 2,Skewed Traffic Run*

	5 Hosts		*10 Hosts*	
	Skewed	*Non-Skewed*	*Skewed*	*Non-Skewed*
1.	36557	(37639)	179068	(183852)
2.	(38819)	36833	326597	(330553)
3.	47474	(48195)	(197312)	186395
4.	(12213)	11192	(76878)	75689

Table 4.3: *The Initial Topology Cost (The Bigger Cost is in Brackets)*

most likely dependent on the fractions of each pipe. For example, consider the following two sets of pipes $S_1 = \{1.0, 1.0, 1.0, 1.0, 1.0\}, S_2 = \{0.1, 0.1, 0.1, 0.1, 4.6\}$. Both sets have the same mean (1.0). The values in Set S_1 are identical (skewness 0), hence it has lower a skewness than Set S_2. The number of transmission lines needed for Set S_1 will be the summation of the ceiling of each pipe value, that is 5 lines. However, the number of transmission lines needed for Set S_2 is 9 lines. The more transmission lines needed the bigger the initial topology will most likely turn out to be, since more switch expansions need to be done (For example, if the maximum switch size is 8, then Set S_1 will be able to fit into one switch while Set S_2 will not). This is an example where a set of pipes with lower skewness results in a cheaper topology.

On the other hand, if we have a set of pipes, Set $S_3 = \{3.0, 0.5, 0.5, 0.5, 0.5\}$ and Set $S_4 = \{1.1, 1.1, 1.1, 1.1, 0.6\}$. Again, both of them have the same mean (1.0). However, S_3 with a higher skewness (the *minimum* difference of the individual pipes with the mean is 0.5 for Set S_3, whereas the *maximum* difference between the individual pipes and the mean for Set S_4 is only 0.4) requires 7 lines, whereas S_4 with a lower skewness requires a total of 9 lines. This is an example where the set of pipes with a higher skewness results in a cheaper topology. This is the opposite of the results that we get from the previous example.

4.7.3 Part II: Topology Improvement

In the experimental runs, we have examined the performance of the three topology improvement algorithms. Figures 19 to 21 show an example of the results obtained from applying these algorithms to the initial topology in Figure 18. From the

figures, we can see that reduction in topology sizes have been achieved using the topology improvement algorithms. Both the Random and Exhaustive algorithms have managed to reduce the number of switches in the initial topology from a total of 11 to 6, whereas the Neighborhood algorithm managed to reduce it down to just 5 switches (Figure 21).

4.7.4 Repeated Calculation

Due to the fact that the both the Random and Neighborhood topology improvement algorithms depend on random selections of switches to combine, the resultant topologies generated with these two algorithms will vary from one run to another. Therefore to more accurately determine the performance of these two algorithms, we have applied the same algorithm a total of 70 times for each run. The mean and the 95% confidence interval of the topology cost are then obtained and plotted.

Figures 22 to 25 shows some of the plots that we have obtained. They are for the 5-host runs and 15-host runs. Since the results of the 10-host runs are identical to the 5-host runs, we chose not to include them. As evident from the figures, most of the time the Exhaustive algorithm produces the cheapest topology, followed by the Neighborhood algorithm, then the Random algorithm. This seems to confirm that the 'intelligence' that is built into the Exhaustive and Neighborhood does generate good topologies. Note that since the results in Figures 22 to 25 show that the confidence intervals of the bars in the same column do not overlap, we can be fairly sure that the differences are significant.

4.7.5 Part III: Topology Location

Like Part II, both of the topology location algorithms rely on random numbers to work. Therefore, we have run each Topology Location Algorithm 70 times for each input, so that the mean and 95% confidence interval can be obtained.

From the figures, we have found that in all cases the Anchoring algorithm produces a topology that is less costly than that produced by the Random topology location algorithm. This difference in performance can sometimes be as high as

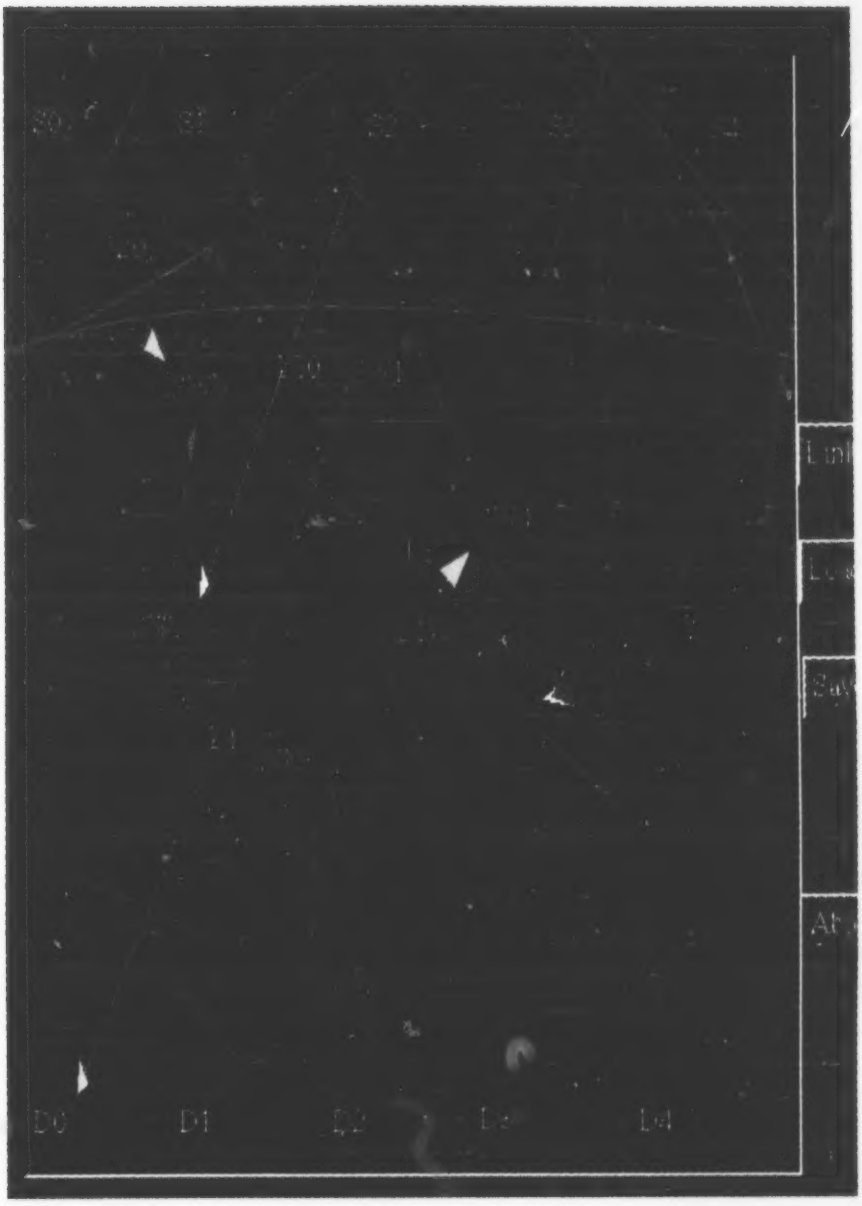

Figure 19: *The Improved Topology Using the Random Algorithm*

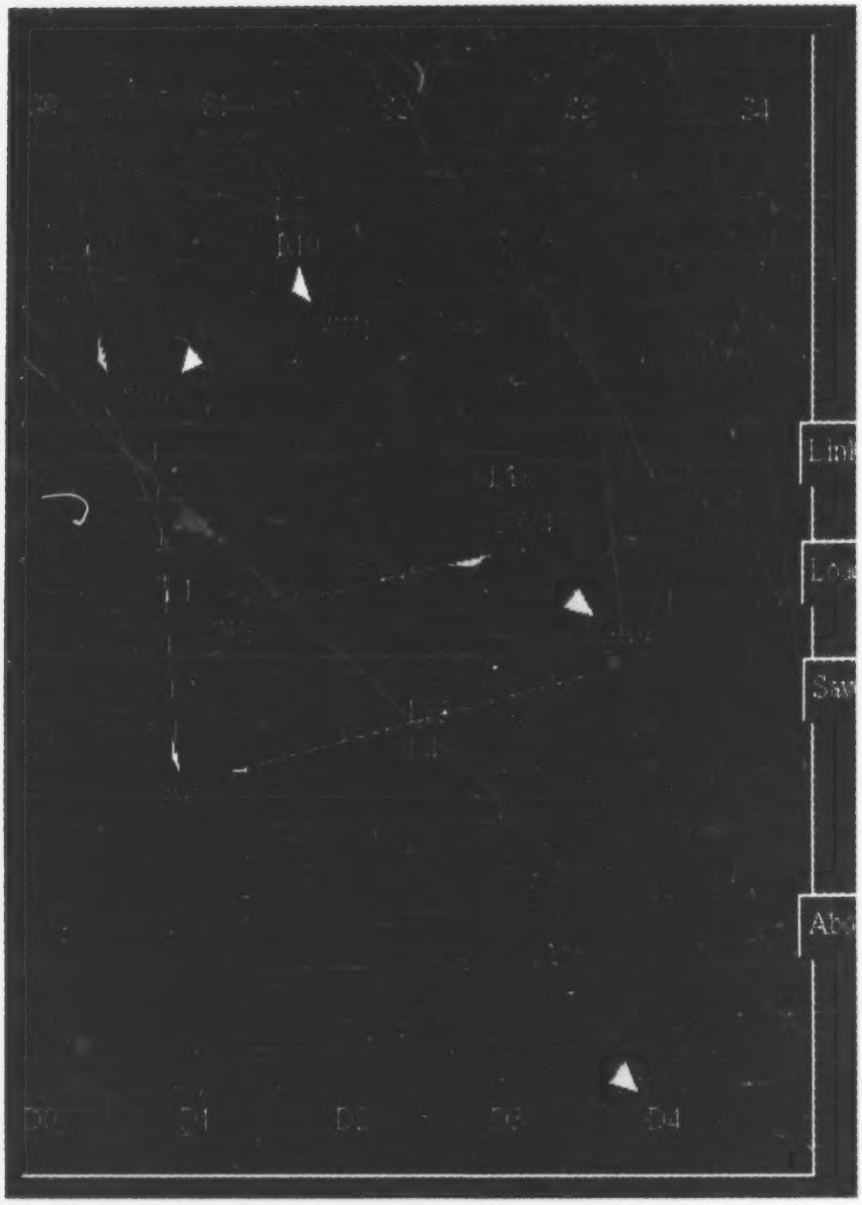

Figure 20: *The Improved Topology Using the Exhaustive Algorithm*

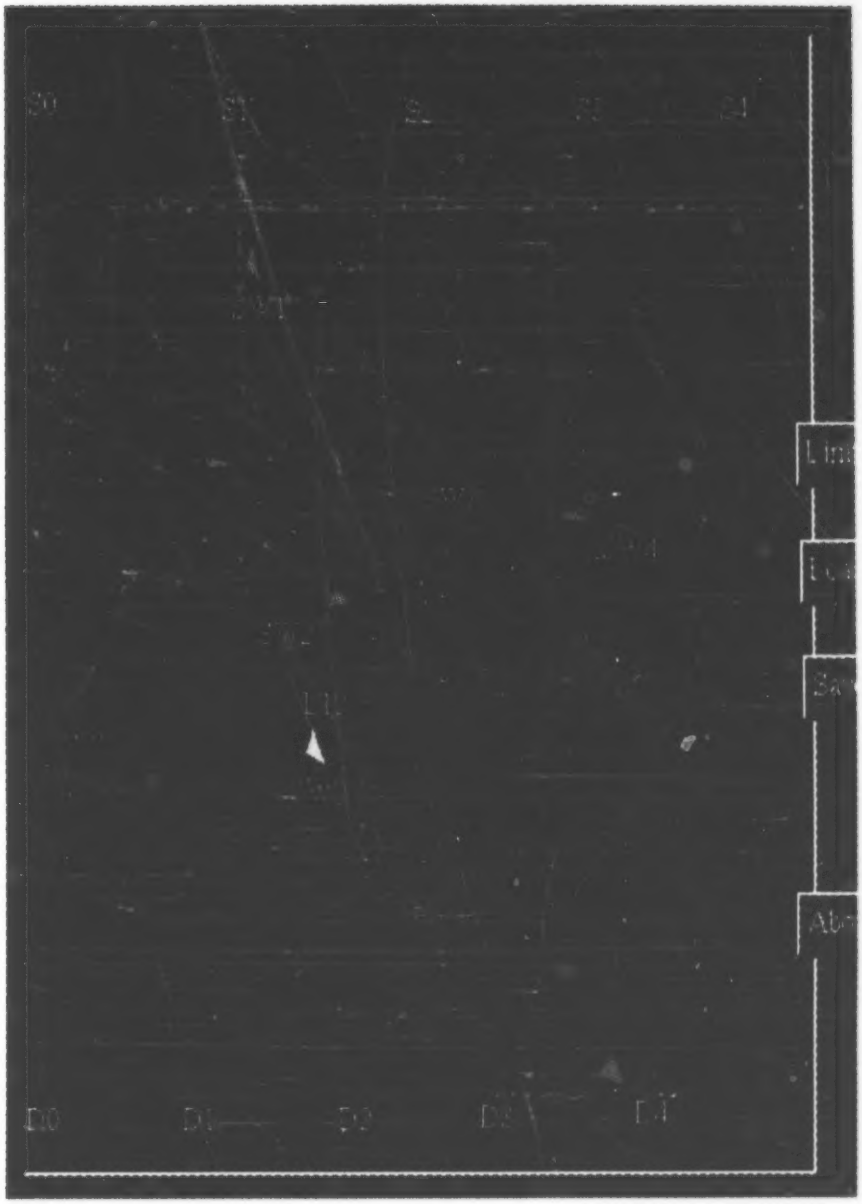

Figure 21: *The Improved Topology Using the Neighborhood Algorithm*

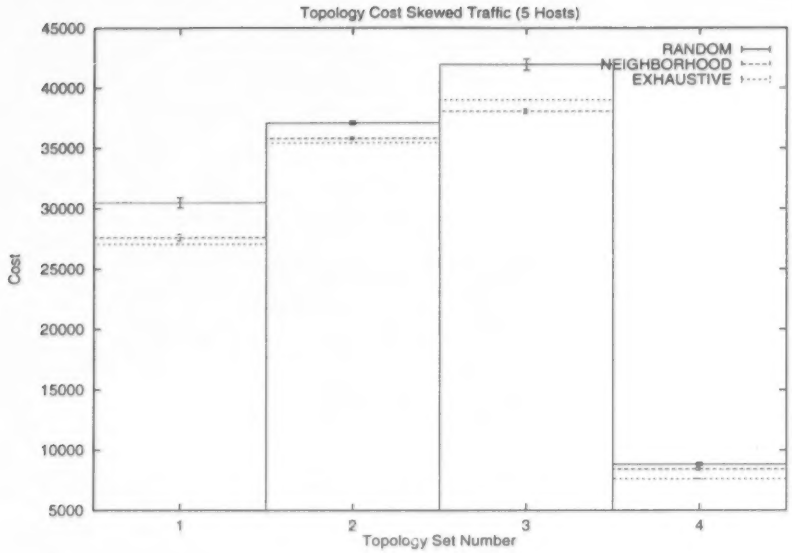

Figure 22: *Comparisons of the Topology Improvement algorithms for a 5 Host Network with Skewed Traffic*

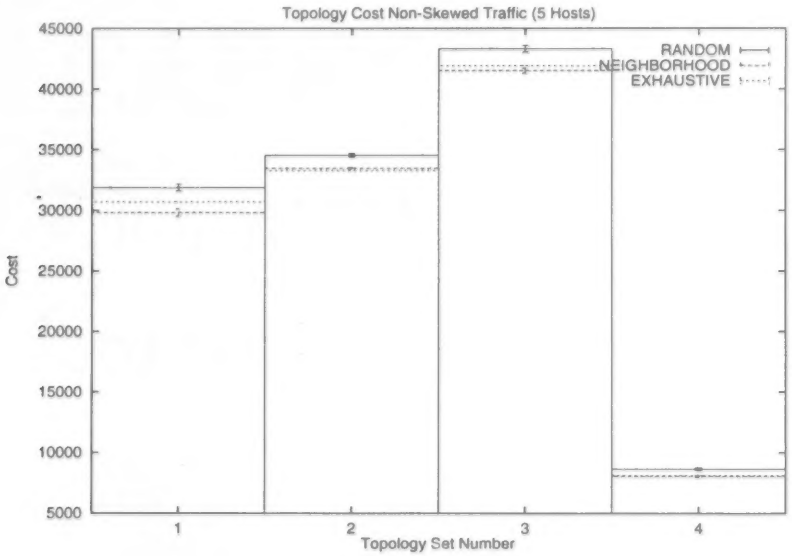

Figure 23: *Comparisons of the Topology Improvement Algorithms for a 5 Host Network with Non-Skewed Traffic*

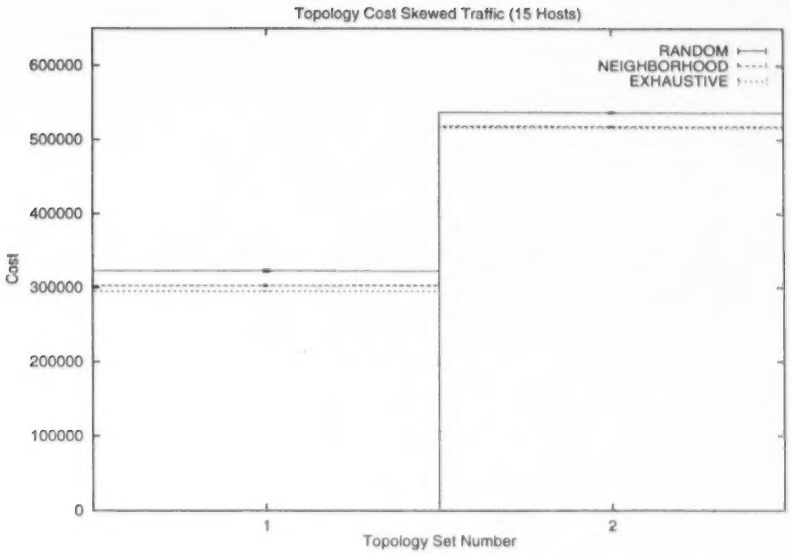

Figure 24: *Comparisons of the Topology Improvement Algorithms for a 15 Host Network with Skewed Traffic*

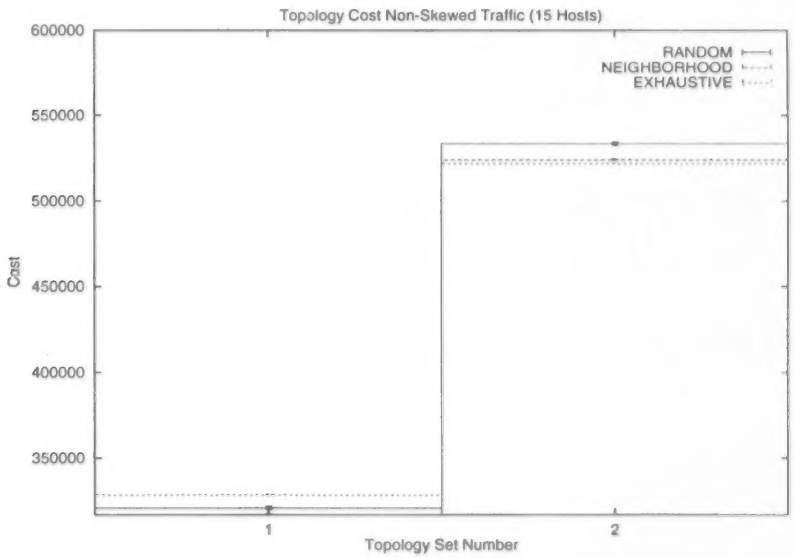

Figure 25: *Comparisons of the Topology Improvement Algorithms for a 15 Host Network with Non-Skewed Traffic*

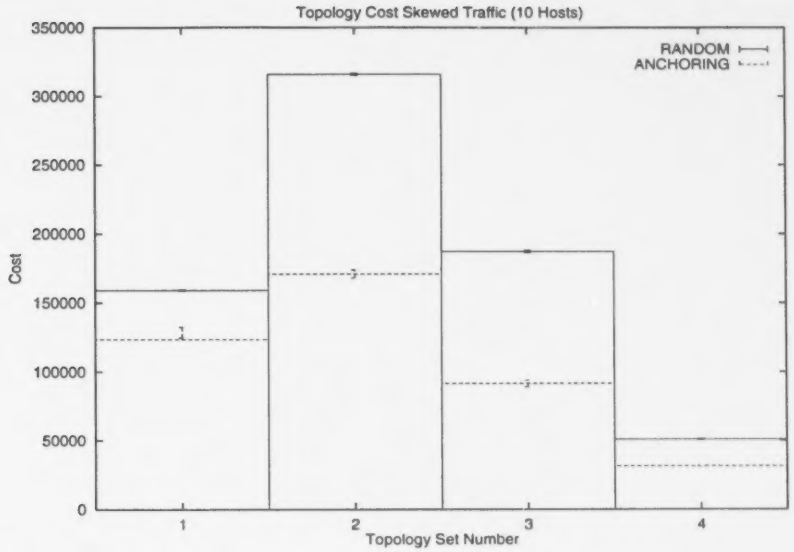

Figure 26: *Comparisons of the Topology Location Algorithms for a 10 Host Network with Skewed Traffic*

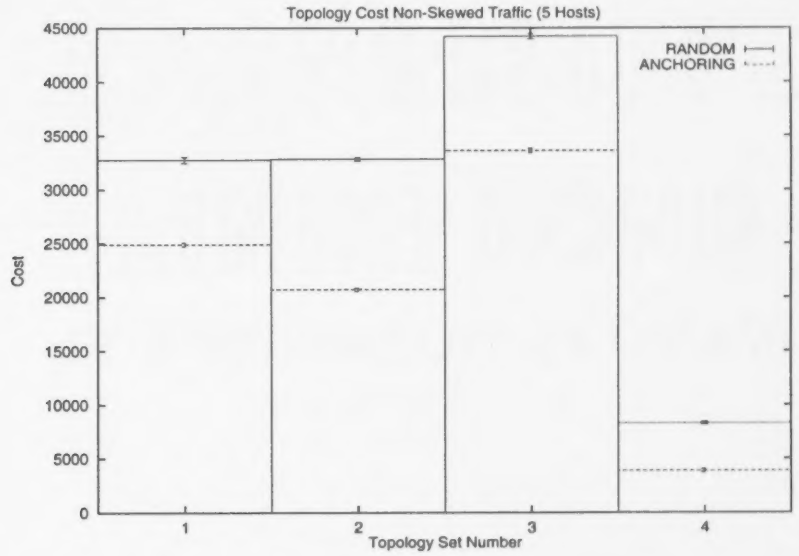

Figure 27: *Comparisons of the Topology Location Algorithms for a 5 Host Network with Non-Skewed Traffic*

two times (Figure 27). This is not surprising at all, especially when some of the switch candidate locations are very far from the host. In the Anchoring algorithm, these locations will be avoided. However, for the Random algorithm, these switch candidate locations can still be chosen as the place to put switches.

4.7.6 Algorithm Running Time

Although the Exhaustive algorithm emerge as the winner in Part II, it is not without its downside. The major problem with this algorithm is its exponential increase in execution time with the rise in number of hosts in the network. Figure 28 shows the execution time for the Part II algorithms, and 29 shows the execution time for the Part III algorithms. These are real time not CPU time calculations and may not be completely accurate indicators of the execution speed since other processes may be present on the system. However, the results are consistent over the many runs and the differences in execution time for the different algorithms are large.

As we can see from Figure 28, for a 20 host case the execution time for the Exhaustive algorithm goes up to 8606 seconds, which is about 2.3 hours. These are real time measures and therefore, the differences are consistently significant. Thus, for large host size, for example 100 hosts, the execution time for the Exhaustive algorithm may become too long to be of any use. Nevertheless, since this method is specifically designed for local area networks, chances are that the number of hosts in the network will not reach that high.

For the Anchoring algorithm, even though it is also showing an exponential increase in execution time with the increase of host number, the actual time that it takes to run is very small (in fractions of a second even for 20 hosts). Hence, we believe that long running time will not be the problem for larger hosts numbers.

4.7.7 Compatibility of Part II and Part III

Table 4.4 shows the results of some of the runs, before and after applying the topology location Anchoring algorithm to them.

We can see that in some runs, even though the Random topology improvement

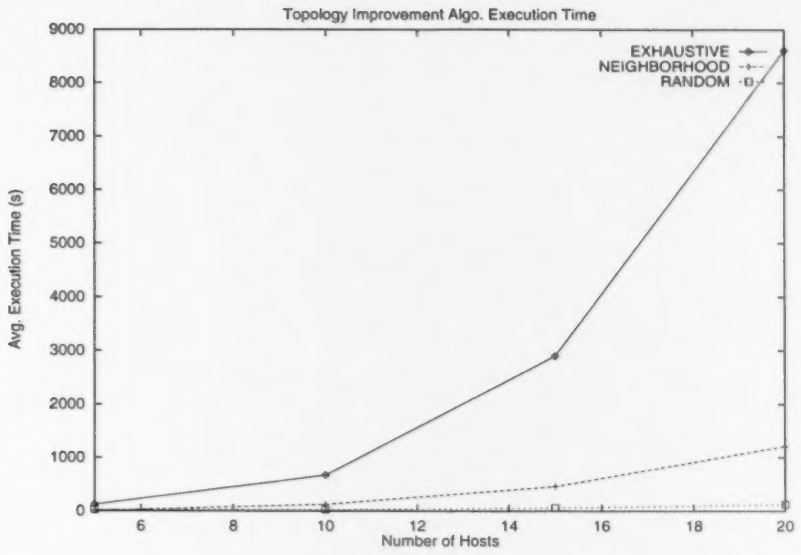

Figure 28: *Average Execution Time for the Topology Improvement Algorithms (in Seconds)*

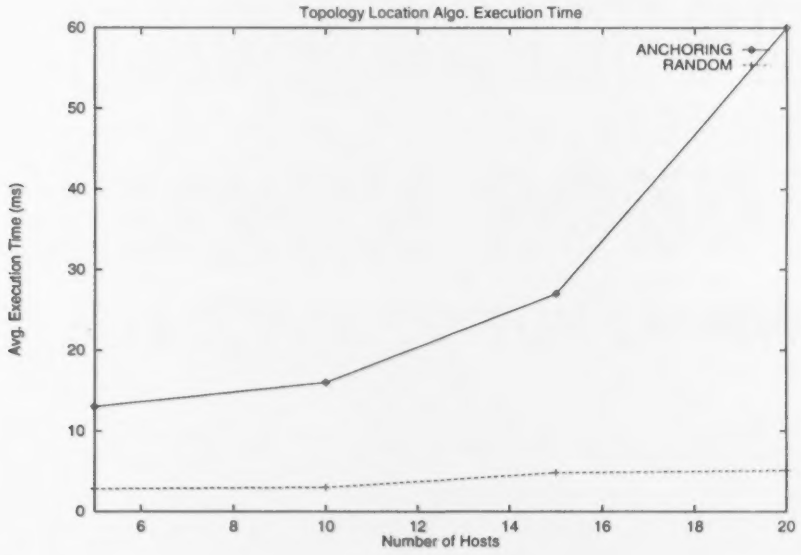

Figure 29: *Average Execution Time for the Topology Location Algorithms (in Milliseconds)*

	Random		*Exhaustive*		*Neighborhood*	
	Before	After	Before	After	Before	After
1.	162235	123292	156210	127473	160164	124131
2.	317830	171938	315964	176189	315088	176699
3.	174901	87669	167203	97435	170006	89114
4.	49336	30867	47636	30808	48343	30825

Table 4.4: *Results of the Three Topology Improvement Algorithms Before and After Anchoring*

algorithm loses in the beginning, after applying the anchoring algorithm, it reverses and becomes the winner (Sets 1 to 3). This is an interesting observation, as we thought that 'minimizing' the cost in Parts II and III separately should yield good results. However, as we have seen, this is not always true. This implies that it would be better to combine both Part II and Part III together so that we can take into account both switch locations and switch combinations at the same time. Another view is that there is no sense using "intelligence" at the topology improvement stage; just choose something quick.

4.8 Summary

In this chapter, Phase II of our Pipe-Based Topology Design Methodology is presented in detail. The results of the experiments that were run to analyze the algorithms in Phase II are also presented. From the results we can conclude that:

- The Initial Topology algorithm successfully generates a feasible topology which the Part II topology Improvement algorithms will improve.
- The algorithms proposed in Part II successfully reduces the size of the initial topology by combining switches.
- The Exhaustive algorithm in overall exhibits the most savings at the Topology Improvement Stage, followed by the Neighborhood algorithm, then by the Random algorithm.

- However, the Exhaustive algorithm has the longest execution time, followed by the Neighborhood algorithm, then by the Random algorithm.
- The Anchoring Location algorithm assigns switches to better locations than the Random Location algorithm in all cases.
- The separation of Part II and Part III will not always give good results. For future improvements, Part II and Part III should be combined.

Chapter 5

Conclusion

5.1 Summary

In this thesis, we have proposed a novel method for building topologies for local area ATM networks (LATM). This method utilizes our concept of ATM pipes as the fundamental building blocks of the network. In determining the size of these pipes, we have developed a mathematical method (the Multiclass Heterogeneous On/Off Sources Fluid-Flow Method, MCFF) by extending on the work of Tucker [39]. Then, we proposed a 3 part methodology to construct the LATM network. We also introduced and compared 6 variations of the implementation of this methodology.

We started with Chapter 1, where we presented a general discussion on topology design. Then in Chapter 2, an overview of some techniques on topology from previous literature was given.

In Chapter 3, we presented an overview to our methodology for building a LATM network. Basically, there are two phase to our method. The first phase involves determining the pipe sizes required, whereas the second phase concerns with building the topology based on the calculated pipe sizes. The first phase of our method, i.e. the MCFF model was presented in detail. To verify the accuracy of the MCFF, results from some simulation runs that we have performed was presented. If was found that the MCFF is very accurate, since the values calculated with MCFF lie within the 95% confidence interval of their corresponding

simulation runs.

The second phase of our method was presented in Chapter 4. Each of the proposed algorithms was outlined, and performance of the algorithms are given. It was found that for topology improvement, the best algorithm is the Exhaustive algorithm, followed by the Neighborhood algorithm, then the Random algorithm. For topology location, Anchoring performs better than Random topology location in all cases. However, overall when Part 3 is applied it may reverse the result of Part 2, and hence may mean that we should not separate the 2 functions. Other results are, skewness of pipe sizes has little effect on the results, and that the execution times of the algorithms are significantly different.

Conclusions and future work are then summed up in this Chapter.

5.2 Conclusions and Future Work

In this section we conclude our method successfully generates LATM topologies while satisfying the QOS (specifically, the traffic loss probability) requirements. One of the advantages of our work is that the method is highly modular. That is if a better solution exists for the parts, we can simply substitute the new solution into that part. Our method is also practical, since pipes can be implemented as virtual paths. Another conclusion is that the MCFF model is very accurate. However, improvements can still be done to model other types of traffic as well. A disadvantage of our method is that since the topology construction phase is further divided into three separate parts, problems of incompatibility between the parts may occur. We have seen this with the overall cost of the finished network. One algorithm at the topology improvement stage may be the best but is poorer at the final stage.

Possible future work for this thesis are

- *Pipe Calculation* Currently, we assume that the traffic in the pipes are fixed rate. Research can be done, so that other traffic types, such as variable bit-rate traffic (VBR) such as VBR voice, compress video can be taken into account. At this point, this traffic would be handled by this model if peak

rate is applied. Another possible research work is modelling the effects of cell scale as well as burst scale. MCFF is currently a burst scale model.

- *Pipe Usage* This involves investigating the effects of grouping certain types of traffic together, or the usage of pipes. In theory, a source can send traffic to the destination in more than one pipe. By having multiple pipes for a single source to the destination, essentially that means that a source can have a choice of different loss rates to send its traffic. Another good item for investigation is the idea of a broadcast pipe. This is in anticipation of the multicast or broadcast traffic on the network.

- *Pipe Configuration* After a network has been installed, the traffic that runs on the network changes through time. The pipe configuration that was initially done at the time of installation may not apply anymore. Research can be done to see how a new pipe configuration can be derived from the underlying switched network without changing the physical topology.

- *Physical Topology Redesigning* Research can be carried out in this area to determine how a new physical topology using ATM pipes can be added on a existing ATM pipe network with little or no change to the existing topology.

Appendix A

Tucker's Homogeneous On/Off Sources Fluid-Flow Model

This appendix presents Tucker's Fluid-Flow Model for multiplexed homogeneous On/Off voice sources, on which the model presented in this thesis for the calculation of a multiclass, heterogeneous On/Off sources is extended in Chapter 3.

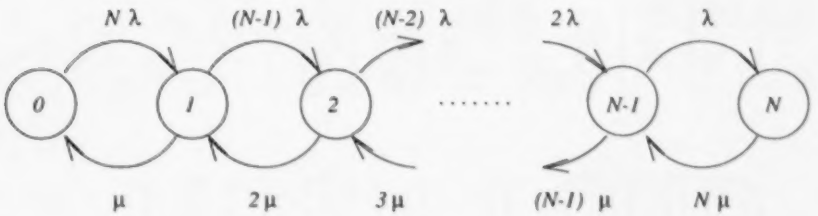

Figure 1: *The Talker Activity Model Markov Chain*

The Fluid-Flow Model is based on a Markov Chain, as shown in figure 1, This Markov Chain is also known as the Talker Activity Model.

The state of the Markov Chain, is represented by the number in the circle denotes the the number of On/Off sources in the On state out of a total of N sources. Each source is a voice source that alternates between an identical exponentially distributed On period (talkspurt), where voice data is generated and an exponentially distributed Off period, where no voice data is produced

(silence). These On/Off periods has a mean of

$$\begin{aligned} 1/\mu &= \text{mean talkspurt length} \\ 1/\lambda &= \text{mean silence length} \end{aligned}$$

With the exponentially distributed On/Off periods, the birth and death rates for the Talker Activity Model are then

$$\begin{aligned} p(i,i+1) &= (N-i)\lambda, i \neq N \\ p(i,i-1) &= i\mu, i \neq 0 \end{aligned}$$

where $p(i,j)$ denotes the transition rate from state i, where i sources are in their On periods, and to state j, where j sources are on. The total flow-rate out of state i is then

$$p^*(i) = p(i,i+1) + p(i,i-1) = (N-i)\lambda + i\mu$$

The multiplexed data produced by these N voice sources, after being fed into a FIFO buffer, are then transmitted on to a link with capacity c units. Since Tucker assumes that each voice source generates voice data at a capacity of 1 unit, the contents in the buffer can be observed as follows:

1. If the number of On sources i is less than c $(i < c)$, the queue decreases steadily at a rate of $(c-i)$. If the queue becomes empty, it will remain empty.

2. If $(i > c)$, the queue increases steadily at a rate of $(i-c)$. If the queue reaches its limit, it will stay on its limit.

3. If $(i = c)$, the queue length does not change.

Let $P_i(t,x), 0 \leq i \leq N, 0 \leq t, 0 \leq x \leq m$ be the probability that i sources are on at time t, and the queue length in the buffer is $\leq x$, and that the queue limit is m. Then

$$\begin{aligned} P_i(t+\Delta t, x) &= P_{i-1}\{t, x-(i-c)\Delta t\}p(i-1,i)\Delta t \\ &\quad + P_{i+1}\{t, x-(i-c)\Delta t\}p(i+1,i)\Delta t \\ &\quad + P_i\{t, x-(i-c)\Delta t\}(1-p^*(i)\Delta t) + o(t) \qquad \text{(A.1)} \end{aligned}$$

Dividing both sides by $\triangle t$ and taking $\triangle t \to 0$, with a little manipulation equation (A.1) becomes

$$\delta P_i(t,x)/\delta t + (i-c)\delta P_i(t,x)/\delta x = p(i-1,1)P_{i-1}(t,x) + p(i+1,i)P_{i+1}(t,x) - p^*(i)P_i(t,x), \quad 0 < x < m. \tag{A.2}$$

If we let $F_i(x) \triangleq$ the equilibrium probability ($\lim_{t\to\infty} P_i(t,x), \delta P_i/\delta t = 0$) there are i on sources and the queue length $\leq x$, equation (A.2) can be rewritten as

$$(i-c)\, dF_i/dx = p(i-1,i)F_{i-1}(x) + p(i+1,i)F_{i+1}(x) - p^*(i)F_i(x), \quad 0 < x < m. \tag{A.3}$$

Or in matrix form

$$\mathbf{D}\, d\mathbf{F}(x)/dx = \mathbf{M}\mathbf{F}(x), 0 < x < m \tag{A.4}$$

where

$$\mathbf{D} = \begin{bmatrix} -c & & & & \\ & 1-c & & & \\ & & 2-c & & \\ & & & \ddots & \\ & & & & N-c \end{bmatrix}$$

and

$$\mathbf{M} = \begin{bmatrix} -p^*(0) & p(1,0) & & & & & \\ p(0,1) & -p^*(1) & p(2,1) & & & & \\ & p(1,2) & -P^*(2) & p(3,2) & & & \\ & & & \ddots & & & \\ & & & p(N-2,N-1) & -p^*(N-1) & p(N,N-1) \\ & & & & p(N-1,N) & -p*(N) \end{bmatrix}$$

Hence, there are $(N+1)$ state equations to be solved, assuming c is not an integer. If z_r be an eigenvalue of $\mathbf{D}^{-1}\mathbf{M}$ and ϕ_r be its corresponding right eigenvector. The solution to (A.4) becomes

$$\mathbf{F}(x) = \sum_{r=0}^{N} \exp(z_r x) a_r \phi_r, \quad 0 < x < m \tag{A.5}$$

By setting suitable boundary conditions, the coefficients a_r for (A.5) can be obtained. Defining $F_i(m-)$ as $\lim_{x \to m} F_i(x)$, and b_i = the equilibrium probability that i lines are on, that is

$$b_i = \frac{(\lambda/\mu)^i \begin{pmatrix} N \\ i \end{pmatrix}}{(1+\lambda/\mu)^N}$$

the boundary conditions can be set by observing the fact that

1. for $(i > c)$, the buffer contents is always increasing, therefore the probability that the buffer contents will be less than or equals to zero is nil. In other words, $F_i(0) = 0$

2. for $(i < c)$, the buffer contents is always decreasing. Hence, $F_i(m-) = b_i$. Equation (A.4) can then be solved by standard numerical algorithms.

The loss can be calculated by

$$\text{Loss} = (1/(\alpha N)) \sum_{i=C}^{N} (i-c)(b_i - F_i(m-)) \qquad (A.6)$$

where α is the average fraction of lines on $\alpha = \lambda/(\mu+\lambda)$, and $C = \lceil c \rceil$.

Bibliography

[1] D Anick, D Mitra, and M Sondhi. Stochastic theory of a data-handling system with multiple sources. *Bell System Technical Journal*, 61:1253–1257, 1982.

[2] Sara Baase. *Computer Algorithms: Introduction to Design and Analysis.* Addison-Wesley Publishing Company, 1988.

[3] Andrew Campbell, Geoff Coulson, and David Hutchison. A quality of service architecture. *Computer Communication Review*, 24(2):6–27, Apr 1994.

[4] John Daigle and Joseph Langford. Models for analysis of packet voice communications systems. *IEEE Journal of Selected Areas in Communications*, 4(6):847–855, Sep 1986.

[5] M. de Prycker, R. Peschi, and T. Van Landegem. B-ISDN and the OSI protocol reference model. *IEEE Network*, 7(2):10–18, March 1993.

[6] Martin de Prycker. *Asynchronous Transfer Mode: Solution for Broadband ISDN.* Ellis Horwood, 1991.

[7] Anwar Elwalid and Mitra Debasis. Fluid models for the analysis and design of statistical multiplexing with loss priorities on multiple classes of bursty traffic. *IEEE Infocom*, 1:415–425, 1992.

[8] Noboru Endo, Toshiya Ohuchi, Takahiko Kozaki, Hiroshi Kuwahara, and Makoto Mori. Traffic characteristics evaluation of a shared buffer switch. *IEEE Globecom*, 3:1913–1918, 1990.

[9] Cem Ersoy and Shivendra Panwar. Topological design of interconneccted LAN-MAN networks. *IEEE Infocom*, 3:2260–2269, 1992.

[10] Michael Garey and David Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W. H. Freeman and Co., 1979.

[11] Mario Gerla, Jose Augusto Monteiro, and Rodolfo Pazos. Topology design and bandwidth allocation in ATM nets. *IEEE Journal on Selected Areas in Communications*, 7(8):1253–1262, October 1989.

[12] Donald Gross and Carl M. Harris. *Fundamentals of Queueing Theory.* John Wiley and Sons, 1985.

[13] Harry Heffes and David M Lucantoni. A markov modulated characterization of packetized voice and data traffic and related statistical multiplexer performance. *IEEE Journal of Selected Areas in Communications*, 4(6):856–867, September 1986.

[14] Ellis Horowitz and Sartaj Sahni. *Fundamentals of Computer Algorithms.* Computer Science Press, 1978.

[15] W Hsieh, M Gerla, P McGregor, and J. Eckl. Locating backbone switches in a large packet network. In *Proceedings of the 3rd International Conference on Computer Communication*, pages 330–339, Toronto, 1976.

[16] Aaron Kershenbaum. *Telecommunications Network Design Algorithm.* McGraw-Hill, 1993.

[17] Aaron Kershenbaum, Parviz Kermani, and George A. Grover. Mentor: An algorithm for mesh network topological optimization and routing. *IEEE Transactions on Communications*, 39(4):503–513, April 1991.

[18] S Kirkpatrick, C. D. Gelatt, and M Vecchi. Optimization by simulated annealing. *Science*, pages 671–680, May 1983.

[19] Leonard Klienrock. *Queueing Systems: Vol I.* John-Wiley and Sons, 1976.

[20] L Kosten. Stochastic theory of a multi-entry buffer. *Delft Progress Report*, pages 10–18, 1974.

[21] H. Kroner, M. Eberspacher, T. H. Theimer, P Kuhn, and U. Briem. Approximate analysis of the end-to-end delay in ATM networks. *IEEE Infocom*, pages 978–986, 1992.

[22] Hiroshi Kuwahara, Nonoru Endo, Mineo Ogino, and Takahiko Kozaki. A shared buffer memory switch for an ATM exchange. In *Proceedings: Internationl Conference on Communications*, pages 118–122, 1989.

[23] Averill M. Law and Kelton W. David. *Simulation Modeling And Analysis.* McGraw-Hill, Inc., 1991.

[24] Ming-Jing Lee and James R. Yee. A design algorithm for reconfigurable ATM networks. *IEEE Infocom*, 1:144–151, 1993.

[25] Ian Leslie, McAuley R., and David Tennenhouse. ATM everywhere? *IEEE Network*, 7(2):40–46, March 1993.

[26] Ke-Qiang Liao and Lorne Mason. A heuristic approach for performance analysis of ATM systems. *IEEE Globcom*, 3:1931–1935, 1990.

[27] Chu-Kiat Lim and Ursula Maydell. Xpress transfer protocol in a medical communication system. In *(to appear) Proceedings of SPIE, The International Society of Optical Engineering. Medical Imaging V: PACS Design and Evaluation*, Newport Beach, February 1994.

[28] J Bryan Lyles and Daniel C. Swinehart. The emerging gigabit environment and the role of local ATM. *IEEE Communications Magazine*, pages 52–58, Apr 1992.

[29] B. Martos. *Nonlinear Programming Theory and Methods.* Amsterdam and Oxford: North-Holland, 1975.

[30] Claude Mcmillan Jr. *Mathematical Programming: An Introduction to the Design and Application of Optimal Decision Machines.* John Wiley and Sons, Inc., 1970.

[31] Steven E. Minzer. Broadband ISDN and Asynchronous Transfer Mode (ATM). *IEEE Communications Magazine*, pages 17–24, September 1989.

[32] Jose Augusto Monteiro and Mario Gerla. Topological reconfiguration of ATM networks. In *Proceedings of IEEE Infocom*, pages 207–214, San Francisco, California, June 1990.

[33] Yuji Oie, tatsuya Suda, Masayuki Murata, David Kolson, and Hideo Miyahara. Survey of switching techniques in high-speed networks and their performance. In *IEEE Infocom*, pages 1242–1251, 1990.

[34] Tadanobu Okada, Hirokazu Ohnishi, and Naotaka Morita. Traffic control in Asynchronous Transfer Mode. *IEEE Communications Magazime*, pages 58–62, September 1991.

[35] R. Onvural. *Asynchronous Transfer Mode Networks: Performance Issues.* Archtech House, Inc., 1994.

[36] HuanXu Pan, Hiroyuki Okazaki, and Issei Kino. Analysis of a gradual input model for bursty traffic in ATM. *Teletraffic and Datatraffic (IAC)*, pages 795–800, 1991.

[37] Craig Patridge. *Gigabit Networking.* Addison-Wesley Publishing Company, 1994.

[38] Youichi Sato and Ken-Ichi Sato. Virtual path and link capacity design for ATM networks. *IEEE Journal of Selected Areas in Communications*, 9(1):104–109, January 1991.

[39] Roger C. F. Tucker. Accurate method for analysis of a packet-speech multiplexer with limited delay. *IEEE Transactions on Communications*, 36(4):479–483, April 1988.